Part I

Basic Techniques I: Moments and Deviations

Chapter 5. BASIC TECHNIQUES: MOMENTS and DEVIATIONS

Chapter 5. BASIC TECHNIQUES: MOMENTS and DEVIATIONS

In this chapter we present several methods that make use of the second degree moments, variance and standard deviation, for solving various problems related to randomized algorithms.

Chapter 5. BASIC TECHNIQUES: MOMENTS and DEVIATIONS

In this chapter we present several methods that make use of the second degree moments, variance and standard deviation, for solving various problems related to randomized algorithms.

Chapter 5. BASIC TECHNIQUES: MOMENTS and DEVIATIONS

In this chapter we present several methods that make use of the second degree moments, variance and standard deviation, for solving various problems related to randomized algorithms.

We will discuss, in various details, in this chapter also three important problems: Occupancy (Balls-into-Bins) problem,

Chapter 5. BASIC TECHNIQUES: MOMENTS and DEVIATIONS

In this chapter we present several methods that make use of the second degree moments, variance and standard deviation, for solving various problems related to randomized algorithms.

We will discuss, in various details, in this chapter also three important problems:
Occupancy (Balls-into-Bins) problem,
Stable marriage problem

Chapter 5. BASIC TECHNIQUES: MOMENTS and DEVIATIONS

In this chapter we present several methods that make use of the second degree moments, variance and standard deviation, for solving various problems related to randomized algorithms.

We will discuss, in various details, in this chapter also three important problems: Occupancy (Balls-into-Bins) problem, Stable marriage problemand Coupon selection problem that have many applications.

OCCUPANCY (BALLS-INTO-BINS) PROBLEM

OCCUPANCY (BALLS-INTO-BINS) PROBLEM

PROBLEM: Each of m distinguishable objects (balls) is randomly and independently assigned to one of n distinct classes (bins/boxes).

OCCUPANCY (BALLS-INTO-BINS) PROBLEM

PROBLEM: Each of m distinguishable objects (balls) is randomly and independently assigned to one of n distinct classes (bins/boxes). How does the distribution of balls into boxes look like after k assignments?

OCCUPANCY (BALLS-INTO-BINS) PROBLEM

PROBLEM: Each of m distinguishable objects (balls) is randomly and independently assigned to one of n distinct classes (bins/boxes). How does the distribution of balls into boxes look like after k assignments?

Subproblem 1: How many of the boxes will be empty? What is the probability that, for any given k, k boxes will be empty?

OCCUPANCY (BALLS-INTO-BINS) PROBLEM

PROBLEM: Each of m distinguishable objects (balls) is randomly and independently assigned to one of n distinct classes (bins/boxes). How does the distribution of balls into boxes look like after k assignments?

Subproblem 1: How many of the boxes will be empty? What is the probability that, for any given k, k boxes will be empty?

Subproblem 2: What is the maximum number of balls in a box? (What is the probability p_k, for a given k, that maximum number of balls in some box is k?)

OCCUPANCY (BALLS-INTO-BINS) PROBLEM

PROBLEM: Each of m distinguishable objects (balls) is randomly and independently assigned to one of n distinct classes (bins/boxes). How does the distribution of balls into boxes look like after k assignments?

Subproblem 1: How many of the boxes will be empty? What is the probability that, for any given k, k boxes will be empty?

Subproblem 2: What is the maximum number of balls in a box? (What is the probability p_k, for a given k, that maximum number of balls in some box is k?)

Subproblem 3: What is, for a given k, the expected number e_k of boxes with k balls in?

Subproblem 4: What is probability that all balls land in different boxes? (For $n = 365$ and $m < n$ we get so-called birthday problem)

Subproblem 4: What is probability that all balls land in different boxes? (For $n = 365$ and $m < n$ we get so-called birthday problem)

Surprisingly, **these simple probability problems are at the core of the analyses of many randomized algorithms**.

Remainder I. - Mealy's inequality

Remainder I. - Mealy's inequality

For arbitrary events $\xi_1, \xi_2, \ldots \xi_n$

$$Pr\left[\bigcup_{i=1}^{n} \xi_i\right] \leq \sum_{i=1}^{n} Pr(\xi_i)$$

Remainder I. - Mealy's inequality

For arbitrary events $\xi_1, \xi_2, \ldots \xi_n$

$$Pr\left[\bigcup_{i=1}^{n} \xi_i\right] \leq \sum_{i=1}^{n} Pr(\xi_i)$$

Usefulness of this inequality lies in the fact that it makes no assumption about dependencies among events!

Remainder I. - Mealy's inequality

For arbitrary events $\xi_1, \xi_2, \ldots \xi_n$

$$Pr\left[\bigcup_{i=1}^{n} \xi_i\right] \leq \sum_{i=1}^{n} Pr(\xi_i)$$

Usefulness of this inequality lies in the fact that it makes no assumption about dependencies among events!

Therefore, this inequality allows to analyse phenomena with very complicated interactions (without revealing these interactions).

Remainder III. - COMBINATORIAL INEQUALITIES– I.

Remainder III. - COMBINATORIAL INEQUALITIES– I.

Let us now present several combinatorial inequalities that are often used at the analysis of algorithms.

Remainder III. - COMBINATORIAL INEQUALITIES– I.

Let us now present several combinatorial inequalities that are often used at the analysis of algorithms.

$$\binom{n}{k} = \binom{n}{n-k} = \frac{n!}{k!(n-k)!}$$

Remainder III. - COMBINATORIAL INEQUALITIES– I.

Let us now present several combinatorial inequalities that are often used at the analysis of algorithms.

$$\binom{n}{k} = \binom{n}{n-k} = \frac{n!}{k!(n-k)!}$$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

Remainder III. - COMBINATORIAL INEQUALITIES– I.

Let us now present several combinatorial inequalities that are often used at the analysis of algorithms.

$$\binom{n}{k} = \binom{n}{n-k} = \frac{n!}{k!(n-k)!}$$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

$$e^{-x} > 1 - x$$

Remainder III. - COMBINATORIAL INEQUALITIES– I.

Let us now present several combinatorial inequalities that are often used at the analysis of algorithms.

$$\binom{n}{k} = \binom{n}{n-k} = \frac{n!}{k!(n-k)!}$$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

$$e^{-x} > 1 - x$$

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots$$

Remainder III. - COMBINATORIAL INEQUALITIES– I.

Let us now present several combinatorial inequalities that are often used at the analysis of algorithms.

$$\binom{n}{k} = \binom{n}{n-k} = \frac{n!}{k!(n-k)!}$$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

$$e^{-x} > 1 - x$$

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots$$

$$\binom{n}{k} \leq \frac{n^k}{k!}, \quad \binom{n}{k} \leq \left(\frac{ne}{k}\right)^k, \quad \left(\frac{n}{k}\right)^k \leq \binom{n}{k}$$

For large n

$$\binom{n}{k} \sim \frac{n^k}{k!}.$$

Remainder IV. - COMBINATORIAL INEQUALITIES – II

Remainder IV. - COMBINATORIAL INEQUALITIES – II

$$e^t \geq 1 + t \quad \textit{if} \quad t \in \mathbf{R}.$$

Remainder IV. - COMBINATORIAL INEQUALITIES – II

$$e^t \geq 1 + t \quad if \quad t \in \mathbf{R}.$$

If $n \geq 1$ and $|t| \leq n$, then

$$e^t \left(1 - \frac{t^2}{n}\right) \leq \left(1 + \frac{t}{n}\right)^n \leq e^t.$$

Remainder IV. - COMBINATORIAL INEQUALITIES – II

$$e^t \geq 1 + t \quad if \quad t \in \mathbf{R}.$$

If $n \geq 1$ and $|t| \leq n$, then

$$e^t \left(1 - \frac{t^2}{n}\right) \leq \left(1 + \frac{t}{n}\right)^n \leq e^t.$$

For all $t, n \in \mathbf{R}^+$, it holds

$$(1 + \frac{t}{n})^n \leq e^t \leq (1 + \frac{t}{n})^{n+t/2}.$$

Remainder IV. - COMBINATORIAL INEQUALITIES – II

$$e^t \geq 1 + t \quad if \quad t \in \mathbf{R}.$$

If $n \geq 1$ and $|t| \leq n$, then

$$e^t \left(1 - \frac{t^2}{n}\right) \leq \left(1 + \frac{t}{n}\right)^n \leq e^t.$$

For all $t, n \in \mathbf{R}^+$, it holds

$$(1 + \frac{t}{n})^n \leq e^t \leq (1 + \frac{t}{n})^{n+t/2}.$$

nth Harmonic number H_n is defined as follows

$$H_n = \sum_{i=1}^{n} \frac{1}{i} = \ln n + \theta(1).$$

BASIC RESULT for OCCUPANCY PROBLEM

BASIC RESULT for OCCUPANCY PROBLEM

Case: $n = m$

BASIC RESULT for OCCUPANCY PROBLEM

Case: $n = m$

Notation: X_j – the number of balls in the j^{th} bin.

BASIC RESULT for OCCUPANCY PROBLEM

Case: $n = m$
Notation: X_j – the number of balls in the j^{th} bin.
$E[X_i] = 1$ - this can be shown similarly as in case of the *sailor problem*.

BASIC RESULT for OCCUPANCY PROBLEM

Case: $n = m$
Notation: X_j – the number of balls in the j th bin.
$E[X_i] = 1$ - this can be shown similarly as in case of the *sailor problem*.
Notation: $\xi_j(k)$ – the event that bin j has k or more balls in it.

BASIC RESULT for OCCUPANCY PROBLEM

Case: $n = m$
Notation: X_j – the number of balls in the j th bin.
$E[X_i] = 1$ - this can be shown similarly as in case of the *sailor problem*.
Notation: $\xi_j(k)$ – the event that bin j has k or more balls in it.

Analysis of $\xi_1(k)$

BASIC RESULT for OCCUPANCY PROBLEM

Case: $n = m$
Notation: X_j – the number of balls in the $j^{\,th}$ bin.
$E[X_i] = 1$ - this can be shown similarly as in case of the *sailor problem*.
Notation: $\xi_j(k)$ – the event that bin j has k or more balls in it.

Analysis of $\xi_1(k)$
The probability that bin 1 receives exactly i balls is

$$\binom{n}{i}\left(\frac{1}{n}\right)^i\left(1-\frac{1}{n}\right)^{n-i} \leq \binom{n}{i}\left(\frac{1}{n}\right)^i \leq \left(\frac{ne}{i}\right)^i\left(\frac{1}{n}\right)^i = \left(\frac{e}{i}\right)^i$$

BASIC RESULT for OCCUPANCY PROBLEM

Case: $n = m$
Notation: X_j – the number of balls in the $j^{\ th}$ bin.
$E[X_i] = 1$ - this can be shown similarly as in case of the *sailor problem*.
Notation: $\xi_j(k)$ – the event that bin j has k or more balls in it.

Analysis of $\xi_1(k)$
The probability that bin 1 receives exactly i balls is

$$\binom{n}{i}\left(\frac{1}{n}\right)^i\left(1-\frac{1}{n}\right)^{n-i} \leq \binom{n}{i}\left(\frac{1}{n}\right)^i \leq \left(\frac{ne}{i}\right)^i\left(\frac{1}{n}\right)^i = \left(\frac{e}{i}\right)^i$$

Therefore

$$Pr[\xi_1(k)] \leq \sum_{i=k}^{n}\left(\frac{e}{i}\right)^i \leq \left(\frac{e}{k}\right)^k\left(1+\frac{e}{k}+\left(\frac{e}{k}\right)^2+\dots\right)$$

BASIC RESULT for OCCUPANCY PROBLEM

Case: $n = m$
Notation: X_j – the number of balls in the j th bin.
$E[X_i] = 1$ - this can be shown similarly as in case of the *sailor problem*.
Notation: $\xi_j(k)$ – the event that bin j has k or more balls in it.

Analysis of $\xi_1(k)$
The probability that bin 1 receives exactly i balls is

$$\binom{n}{i}\left(\frac{1}{n}\right)^i\left(1-\frac{1}{n}\right)^{n-i} \leq \binom{n}{i}\left(\frac{1}{n}\right)^i \leq \left(\frac{ne}{i}\right)^i\left(\frac{1}{n}\right)^i = \left(\frac{e}{i}\right)^i$$

Therefore

$$Pr[\xi_1(k)] \leq \sum_{i=k}^{n}\left(\frac{e}{i}\right)^i \leq \left(\frac{e}{k}\right)^k\left(1+\frac{e}{k}+\left(\frac{e}{k}\right)^2+\dots\right)$$

and for $k = k^* = \lceil \frac{e \ln n}{\ln \ln n} \rceil$

BASIC RESULT for OCCUPANCY PROBLEM

Case: $n = m$
Notation: X_j – the number of balls in the j th bin.
$E[X_i] = 1$ - this can be shown similarly as in case of the *sailor problem*.
Notation: $\xi_j(k)$ – the event that bin j has k or more balls in it.

Analysis of $\xi_1(k)$
The probability that bin 1 receives exactly i balls is

$$\binom{n}{i}\left(\frac{1}{n}\right)^i\left(1-\frac{1}{n}\right)^{n-i} \leq \binom{n}{i}\left(\frac{1}{n}\right)^i \leq \left(\frac{ne}{i}\right)^i\left(\frac{1}{n}\right)^i = \left(\frac{e}{i}\right)^i$$

Therefore

$$Pr[\xi_1(k)] \leq \sum_{i=k}^{n}\left(\frac{e}{i}\right)^i \leq \left(\frac{e}{k}\right)^k\left(1+\frac{e}{k}+\left(\frac{e}{k}\right)^2+\dots\right)$$

and for $k = k^* = \lceil \frac{e \ln n}{\ln \ln n} \rceil$

$$Pr[\xi_1(k^*)] \leq \left(\frac{e}{k^*}\right)^{k^*}\frac{1}{1-\frac{e}{k^*}} \leq n^{-2}.$$

BASIC COROLLARIES

Problem: What is the probability that at least one bin has at least k^* balls in it?

BASIC COROLLARIES

Problem: What is the probability that at least one bin has at least k^* balls in it?

Solution: It holds

$$Pr\left[\bigcup_{i=1}^{n} \xi_i(k^*)\right] \leq \sum_{i=1}^{n} Pr[\xi_i(k^*)] \leq \frac{1}{n}.$$

BASIC COROLLARIES

Problem: What is the probability that at least one bin has at least k^* balls in it?

Solution: It holds

$$Pr\left[\bigcup_{i=1}^{n} \xi_i(k^*)\right] \leq \sum_{i=1}^{n} Pr[\xi_i(k^*)] \leq \frac{1}{n}.$$

Corollary With the probability at least $1 - \frac{1}{n}$ no bin has more than

$$k^* = \frac{e \ln n}{\ln \ln n}$$

balls in it.

COMPLEXITY of SORTING

COMPLEXITY of SORTING

It is well known that for sorting n elements:

COMPLEXITY of SORTING

It is well known that for sorting n elements:

- Worst case complexity is

COMPLEXITY of SORTING

It is well known that for sorting n elements:

- Worst case complexity is $\mathcal{O}(n \lg n)$;

COMPLEXITY of SORTING

It is well known that for sorting n elements:

- Worst case complexity is $\mathcal{O}(n \lg n)$;
- Average case complexity is

COMPLEXITY of SORTING

It is well known that for sorting n elements:

- Worst case complexity is $\mathcal{O}(n \lg n)$;
- Average case complexity is $\mathcal{O}(n \lg n)$.

COMPLEXITY of SORTING

It is well known that for sorting n elements:

- Worst case complexity is $\mathcal{O}(n \lg n)$;
- Average case complexity is $\mathcal{O}(n \lg n)$.

Both bounds are with respect to the number of comparisons.

COMPLEXITY of SORTING

It is well known that for sorting n elements:

- Worst case complexity is $\mathcal{O}(n \lg n)$;
- Average case complexity is $\mathcal{O}(n \lg n)$.

Both bounds are with respect to the number of comparisons. Can we do better?

COMPLEXITY of SORTING

It is well known that for sorting n elements:

- Worst case complexity is $\mathcal{O}(n \lg n)$;
- Average case complexity is $\mathcal{O}(n \lg n)$.

Both bounds are with respect to the number of comparisons. Can we do better? In some reasonable sense?

COMPLEXITY of SORTING

It is well known that for sorting n elements:

- Worst case complexity is $\mathcal{O}(n \lg n)$;
- Average case complexity is $\mathcal{O}(n \lg n)$.

Both bounds are with respect to the number of comparisons. Can we do better? In some reasonable sense? In some interesting cases?

Reminder - Binomial distribution

Reminder - Binomial distribution

Let now values of a random variable Y be the number of successes of trials with success probability p in n trials.
Then

Reminder - Binomial distribution

Let now values of a random variable Y be the number of successes of trials with success probability p in n trials. Then

$$Pr(Y = k) = \binom{n}{k} p^k q^{n-k}$$

Reminder - Binomial distribution

Let now values of a random variable Y be the number of successes of trials with success probability p in n trials. Then

$$Pr(Y = k) = \binom{n}{k} p^k q^{n-k}$$

Such a probability distribution is called the **binomial distribution** and it holds

Reminder - Binomial distribution

Let now values of a random variable Y be the number of successes of trials with success probability p in n trials. Then

$$Pr(Y = k) = \binom{n}{k} p^k q^{n-k}$$

Such a probability distribution is called the **binomial distribution** and it holds

$$\mathbf{E}Y = np \qquad \mathbf{V}Y = npq \qquad G(z) = (q + pz)^n$$

Reminder - Binomial distribution

Let now values of a random variable Y be the number of successes of trials with success probability p in n trials. Then

$$Pr(Y = k) = \binom{n}{k} p^k q^{n-k}$$

Such a probability distribution is called the **binomial distribution** and it holds

$$\mathbf{E}Y = np \qquad \mathbf{V}Y = npq \qquad G(z) = (q + pz)^n$$

and also

$$\mathbf{E}Y^2 = n(n-1)p^2 + np$$

BUCKET SORT

BUCKET SORT

- Bucket sort is a deterministic sorting algorithm

BUCKET SORT

- Bucket sort is a deterministic sorting algorithm that, under certain probabilistic assumptions on inputs,

BUCKET SORT

- Bucket sort is a deterministic sorting algorithm that, under certain probabilistic assumptions on inputs, sorts numbers in the expected linear time.

BUCKET SORT

- Bucket sort is a deterministic sorting algorithm that, under certain probabilistic assumptions on inputs, sorts numbers in the expected linear time.
- Suppose that we have a set of $n = 2^m$ integers that are to be sorted

BUCKET SORT

- Bucket sort is a deterministic sorting algorithm that, under certain probabilistic assumptions on inputs, sorts numbers in the expected linear time.
- Suppose that we have a set of $n = 2^m$ integers that are to be sorted and they are chosen independently and uniformly at random from the interval $[0, 2^k)$ for a $k \geq m$.

BUCKET SORT

- Bucket sort is a deterministic sorting algorithm that, under certain probabilistic assumptions on inputs, sorts numbers in the expected linear time.
- Suppose that we have a set of $n = 2^m$ integers that are to be sorted and they are chosen independently and uniformly at random from the interval $[0, 2^k)$ for a $k \geq m$.
- Using Bucket sort we can sort such numbers in the expected time $\mathcal{O}(n)$.

BUCKET SORT ALGORITHM - STAGE 1

BUCKET SORT ALGORITHM - STAGE 1

Stage 1. All to be sorted numbers will be put into n buckets

BUCKET SORT ALGORITHM - STAGE 1

Stage 1. All to be sorted numbers will be put into n buckets in such a way that all numbers whose first m bits represent a number j will go to the j-th bucket.

BUCKET SORT ALGORITHM - STAGE 1

Stage 1. All to be sorted numbers will be put into n buckets in such a way that all numbers whose first m bits represent a number j will go to the j-th bucket.

As a consequence, when $j < l$ all elements in the j-th bucket comes before all elements in the l-bucket once all elements are sorted.

BUCKET SORT ALGORITHM - STAGE 1

Stage 1. All to be sorted numbers will be put into n buckets in such a way that all numbers whose first m bits represent a number j will go to the j-th bucket.

As a consequence, when $j < l$ all elements in the j-th bucket comes before all elements in the l-bucket once all elements are sorted.

If we assume that each element can be put in the appropriate bucket in constant time, the above stage requires $\mathcal{O}(n)$ time.

BUCKET SORT ALGORITHM - STAGE 1

Stage 1. All to be sorted numbers will be put into n buckets in such a way that all numbers whose first m bits represent a number j will go to the j-th bucket.

As a consequence, when $j < l$ all elements in the j-th bucket comes before all elements in the l-bucket once all elements are sorted.

If we assume that each element can be put in the appropriate bucket in constant time, the above stage requires $\mathcal{O}(n)$ time.

Because of the assumption that the elements to be sorted are chosen uniformly, the number of elements that land uniformly in a bucket follows the binomial distribution $B(n, \frac{1}{n})$ introduced in Chapter 3.

BUCKET SORT ALGORITHM - STAGE 2

Sort each bucket using a standard quadratic time algorithm and concatenate all sorted lists

BUCKET SORT ALGORITHM - STAGE 2

Sort each bucket using a standard quadratic time algorithm and concatenate all sorted lists

Analysis; If X_i is the number of elements in the ith bucket then they can be sorted in time cX_i^2 for some constant c.

BUCKET SORT ALGORITHM - STAGE 2

Sort each bucket using a standard quadratic time algorithm and concatenate all sorted lists

Analysis; If X_i is the number of elements in the ith bucket then they can be sorted in time cX_i^2 for some constant c.

The expected time to do this sorting is therefore

$$\mathbf{E}\left[\sum_{j=1}^{n} cX_j^2\right] = c\sum_{j=1}^{n}\mathbf{E}[X_j^2] = cn\mathbf{E}[X_1^2]$$

BUCKET SORT ALGORITHM - STAGE 2

Sort each bucket using a standard quadratic time algorithm and concatenate all sorted lists

Analysis; If X_i is the number of elements in the ith bucket then they can be sorted in time cX_i^2 for some constant c.

The expected time to do this sorting is therefore

$$\mathbf{E}\left[\sum_{j=1}^{n} cX_j^2\right] = c\sum_{j=1}^{n}\mathbf{E}[X_j^2] = cn\mathbf{E}[X_1^2]$$

Since X_i is a binomial random variable $B(n, \frac{1}{n})$, see Chapter 3 we get

BUCKET SORT ALGORITHM - STAGE 2

Sort each bucket using a standard quadratic time algorithm and concatenate all sorted lists

Analysis; If X_i is the number of elements in the ith bucket then they can be sorted in time cX_i^2 for some constant c.

The expected time to do this sorting is therefore

$$\mathbf{E}\left[\sum_{j=1}^{n} cX_j^2\right] = c\sum_{j=1}^{n}\mathbf{E}[X_j^2] = cn\mathbf{E}[X_1^2]$$

Since X_i is a binomial random variable $B(n, \frac{1}{n})$, see Chapter 3 we get

$$\mathbf{E}[X_1^2] = \frac{n(n-1)}{n^2} + 1 = 2 - \frac{1}{n} < 2$$

and therefore the expected time of the bucket sort is at most $2cn$.

BIRTHDAY PARADOX - BASICS

BIRTHDAY PARADOX - BASICS

Let us assume that the birthday of each person in a room is a random day chosen uniformly and independently from a 365-day year.

BIRTHDAY PARADOX - BASICS

Let us assume that the birthday of each person in a room is a random day chosen uniformly and independently from a 365-day year. In such case, for any given integer $k > 0$, the probability that in the room with 365 people there are at least k people having their birthdays in different days is:

$$\bar{p}(n) = 1(1 - \frac{1}{365})(1 - \frac{2}{365})(1 - \frac{3}{365}) \dots (1 - \frac{k-1}{365}) = \prod_{j=1}^{k-1}(1 - \frac{j}{365})$$

BIRTHDAY PARADOX - BASICS

Let us assume that the birthday of each person in a room is a random day chosen uniformly and independently from a 365-day year. In such case, for any given integer $k > 0$, the probability that in the room with 365 people there are at least k people having their birthdays in different days is:

$$\bar{p}(n) = 1(1 - \frac{1}{365})(1 - \frac{2}{365})(1 - \frac{3}{365})\ldots(1 - \frac{k-1}{365}) = \prod_{j=1}^{k-1}(1 - \frac{j}{365})$$

what equals to $k!\binom{365}{k}365^{-k}$.

BIRTHDAY PARADOX - BASICS

Let us assume that the birthday of each person in a room is a random day chosen uniformly and independently from a 365-day year. In such case, for any given integer $k > 0$, the probability that in the room with 365 people there are at least k people having their birthdays in different days is:

$$\bar{p}(n) = 1(1-\frac{1}{365})(1-\frac{2}{365})(1-\frac{3}{365})\ldots(1-\frac{k-1}{365}) = \prod_{j=1}^{k-1}(1-\frac{j}{365})$$

what equals to $k!\binom{365}{k}365^{-k}$. Using the inequality $1-\frac{j}{n} \approx e^{-j/n}$ for j small - comparing to n - we have for any integer n

BIRTHDAY PARADOX - BASICS

Let us assume that the birthday of each person in a room is a random day chosen uniformly and independently from a 365-day year. In such case, for any given integer $k > 0$, the probability that in the room with 365 people there are at least k people having their birthdays in different days is:

$$\bar{p}(n) = 1(1 - \frac{1}{365})(1 - \frac{2}{365})(1 - \frac{3}{365})\dots(1 - \frac{k-1}{365}) = \prod_{j=1}^{k-1}(1 - \frac{j}{365})$$

what equals to $k!\binom{365}{k}365^{-k}$. Using the inequality $1 - \frac{j}{n} \approx e^{-j/n}$ for j small - comparing to n - we have for any integer n

$$\prod_{j=1}^{k-1}(1 - \frac{j}{n}) \approx \prod_{j=1}^{k-1} e^{-j/n} = e^{-\sum_{j=1}^{k-1}\frac{j}{n}} = e^{-k(k-1)/2n} \approx e^{-k^2/2n}$$

Hence the probability that some k people in the rom all have different birthdays from a set of n possible birthdays is $\frac{1}{2}$ and it is approximately given by the equation

$$\frac{k^2}{2n} = \ln 2$$

BIRTHDAY PARADOX - BASICS

Let us assume that the birthday of each person in a room is a random day chosen uniformly and independently from a 365-day year. In such case, for any given integer $k > 0$, the probability that in the room with 365 people there are at least k people having their birthdays in different days is:

$$\bar{p}(n) = 1(1 - \frac{1}{365})(1 - \frac{2}{365})(1 - \frac{3}{365})\dots(1 - \frac{k-1}{365}) = \prod_{j=1}^{k-1}(1 - \frac{j}{365})$$

what equals to $k!\binom{365}{k}365^{-k}$. Using the inequality $1 - \frac{j}{n} \approx e^{-j/n}$ for j small - comparing to n - we have for any integer n

$$\prod_{j=1}^{k-1}(1 - \frac{j}{n}) \approx \prod_{j=1}^{k-1} e^{-j/n} = e^{-\sum_{j=1}^{k-1}\frac{j}{n}} = e^{-k(k-1)/2n} \approx e^{-k^2/2n}$$

Hence the probability that some k people in the rom all have different birthdays from a set of n possible birthdays is $\frac{1}{2}$ and it is approximately given by the equation

$$\frac{k^2}{2n} = \ln 2$$

what gives, for the case $n = 365$, $k = 22.49$

$$\prod_{j=1}^{k-1}(1-\frac{j}{365}) = \frac{\prod_{j=1}^{k-1}(365-j)}{365^{k-1}} \cdot \frac{(365-k)!}{(365-k)!}$$

$$= \frac{(365-1)!}{365^{k-1}(365-k)!} \cdot \frac{(365)}{(365)} = \frac{365!}{365^{k}(365-k)!} \cdot \frac{k!}{k!} = k! \binom{365}{k} 365^{-k}.$$

VARIATIONS on BIRTHDAY PARADOX

A more detailed analysis of the basic equation shows that if we have 23 (30) [50] people in one room, then the probability that two of them have the same birthday is more than 50.7%

VARIATIONS on BIRTHDAY PARADOX

A more detailed analysis of the basic equation shows that if we have 23 (30) [50] people in one room, then the probability that two of them have the same birthday is more than 50.7% (70.6%)

VARIATIONS on BIRTHDAY PARADOX

A more detailed analysis of the basic equation shows that if we have 23 (30) [50] people in one room, then the probability that two of them have the same birthday is more than 50.7% (70.6%) (97%) -

VARIATIONS on BIRTHDAY PARADOX

A more detailed analysis of the basic equation shows that if we have 23 (30) [50] people in one room, then the probability that two of them have the same birthday is more than 50.7% (70.6%) (97%) -this is so called the Birthday Paradox.

VARIATIONS on BIRTHDAY PARADOX

A more detailed analysis of the basic equation shows that if we have 23 (30) [50] people in one room, then the probability that two of them have the same birthday is more than 50.7% (70.6%) (97%) -this is so called the Birthday Paradox.

In the case we have 57 [100] people in the room the probability is 99% [99.99997%]

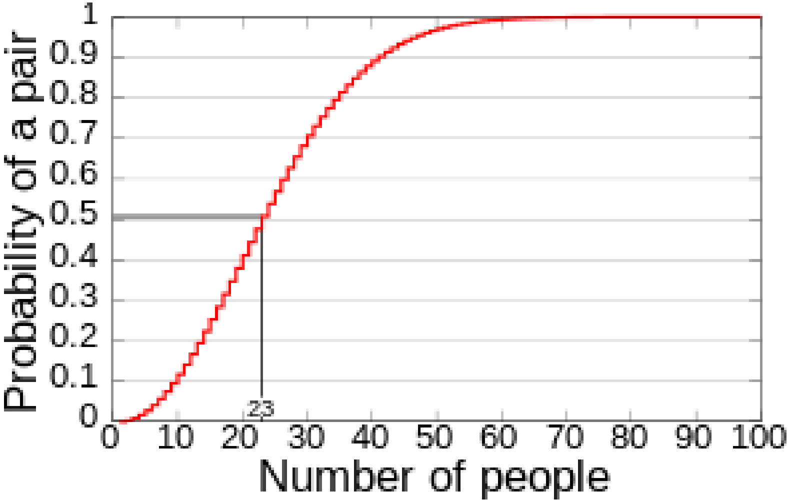
VARIATIONS on BIRTHDAY PARADOX

A more detailed analysis of the basic equation shows that if we have 23 (30) [50] people in one room, then the probability that two of them have the same birthday is more than 50.7% (70.6%) (97%) -this is so called the Birthday Paradox.

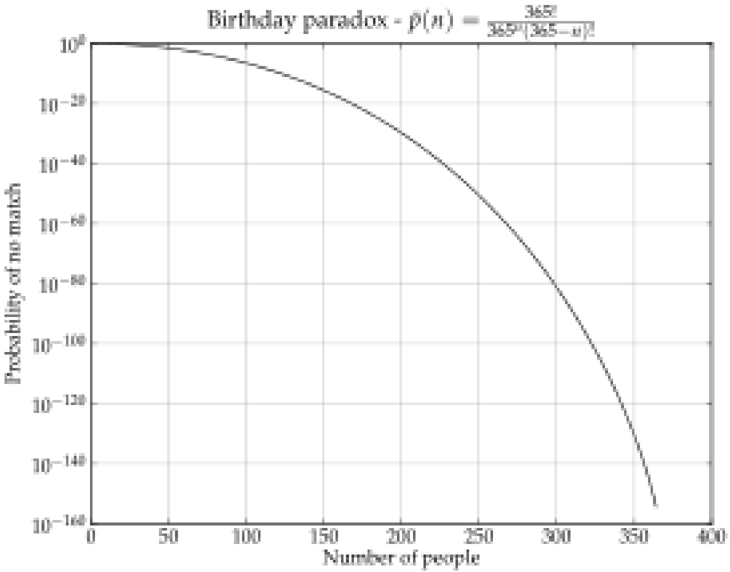
In the case we have 57 [100] people in the room the probability is 99% [99.99997%]

More generally, if we have n objects and r people each choosing one object (and several of them can choose the same object), then if $r \approx 1.177\sqrt{n}$ ($r \approx \sqrt{2\lambda}$), then probability that two people choose the same object is 50% $(1 - e^{\lambda})$%.

Birthday paradox - graph - I.



Birthday paradox - graph - II.



ANOTHER VERSION of THE BIRTHDAY PARADOX

Let us have n objects and two groups of r people. If $r \approx \sqrt{\lambda n}$ then the probability that someone from one group chooses the same object as someone from the other group is $1 - e^{-\lambda}$.

STABLE MARRIAGE PROBLEM - INFORMAL FORMULATION

Given is n men and n women and each of them has ranked all members of the opposite sex with a unique number between 1 and n in order to express of his/her preferences.

STABLE MARRIAGE PROBLEM - INFORMAL FORMULATION

Given is n men and n women and each of them has ranked all members of the opposite sex with a unique number between 1 and n in order to express of his/her preferences.

Task: Marry all men and women together in such a way that there are no two (unsatisfied) people of the opposite sex who would both rather have each other than their current partners.

STABLE MARRIAGE PROBLEM - INFORMAL FORMULATION

Given is n men and n women and each of them has ranked all members of the opposite sex with a unique number between 1 and n in order to express of his/her preferences.

Task: Marry all men and women together in such a way that there are no two (unsatisfied) people of the opposite sex who would both rather have each other than their current partners.

If there is a no dissatisfied couple in a (group) marriage we consider the (group) marriage as stable.

THE STABLE MARRIAGE PROBLEM

Consider a society of n men $A, B, C, \ldots$
and n women $a, b, c, \ldots$

THE STABLE MARRIAGE PROBLEM

Consider a society of n men $A, B, C, \ldots$
and n women $a, b, c, \ldots$

A **marriage** is 1-1 correspondence between men and women of that society.

THE STABLE MARRIAGE PROBLEM

Consider a society of n men $A, B, C, \ldots$
and n women $a, b, c, \ldots$

A **marriage** is 1-1 correspondence between men and women of that society.

Assume that each person has a preference list of the members of the opposite sex, organised in a decreasing order of desirability.

THE STABLE MARRIAGE PROBLEM

Consider a society of n men $A, B, C, \ldots$
and n women $a, b, c, \ldots$

A **marriage** is 1-1 correspondence between men and women of that society.

Assume that each person has a preference list of the members of the opposite sex, organised in a decreasing order of desirability.

Example $A : abcd$ $B : bacd$ $C : adcb$ $D : dcab$
$a : ABCD$ $b : DCBA$ $c : ABCD$ $d : CDAB$

A marriage is said to be **unstable** if there exist two married couples $X - x$, $Y - y$ such that X desires y more than x
y desires X more than Y

THE STABLE MARRIAGE PROBLEM

Consider a society of n men $A, B, C, \ldots$
and n women $a, b, c, \ldots$

A **marriage** is 1-1 correspondence between men and women of that society.

Assume that each person has a preference list of the members of the opposite sex, organised in a decreasing order of desirability.

Example $A : abcd$ $B : bacd$ $C : adcb$ $D : dcab$
$a : ABCD$ $b : DCBA$ $c : ABCD$ $d : CDAB$

A marriage is said to be **unstable** if there exist two married couples $X - x$, $Y - y$ such that X desires y more than x
y desires X more than Y

Such a pair (X, y) is called **dissatisfied**.

THE STABLE MARRIAGE PROBLEM

Consider a society of n men $A, B, C, \ldots$
and n women $a, b, c, \ldots$

A **marriage** is 1-1 correspondence between men and women of that society.

Assume that each person has a preference list of the members of the opposite sex, organised in a decreasing order of desirability.

Example $A : abcd$ $B : bacd$ $C : adcb$ $D : dcab$
$a : ABCD$ $b : DCBA$ $c : ABCD$ $d : CDAB$

A marriage is said to be **unstable** if there exist two married couples $X - x$, $Y - y$ such that X desires y more than x
y desires X more than Y

Such a pair (X, y) is called **dissatisfied**.

The task is to find a **stable marriage**. (At least one always exist!)

Example of an unstable marriage: $A - a$, $B - b$, $C - c$, $D - d$.

COMMENTS

- The stable marriage problem, and its variations, form one of the most famous and important groups of algorithmic problems with a variety of interesting and important applications.

COMMENTS

- The stable marriage problem, and its variations, form one of the most famous and important groups of algorithmic problems with a variety of interesting and important applications.
- A related book: Donald E. Knuth: Stable marriage and its relation to other combinatorial problems: an introduction to the mathematical analysis of algorithms, CRM Proceedings and Lecture Notes,

COMMENTS

- The stable marriage problem, and its variations, form one of the most famous and important groups of algorithmic problems with a variety of interesting and important applications.
- A related book: Donald E. Knuth: Stable marriage and its relation to other combinatorial problems: an introduction to the mathematical analysis of algorithms, CRM Proceedings and Lecture Notes,
- Algorithms to deal with this type of problems are used, for example:
  - To assign graduates of medical schools in North America (about 30 000) to hospitals;

EXISTENCE and OPTIMALITY of SOLUTIONS

NOTE 1: We will show later that a stable marriage always exists.

EXISTENCE and OPTIMALITY of SOLUTIONS

NOTE 1: We will show later that a stable marriage always exists.

NOTE 2: A stable marriage assignment does not need to be optimal for all.

EXISTENCE and OPTIMALITY of SOLUTIONS

NOTE 1: We will show later that a stable marriage always exists.

NOTE 2: A stable marriage assignment does not need to be optimal for all.

EXAMPLE: Let us have three men M_1, M_2 and M_3 and three women W_1, W_2 and W_3 with preferences:

$$M_1 : W_2 W_1 W_3, \quad M_2 : W_3 W_2 W_1, \quad M_3 : W_1 W_3 W_2$$

$$W_1 : M_2 M_1 M_3, \quad W_2 : M_3 M_2 M_1, \quad W_3 : M_1 M_3 M_2$$

EXISTENCE and OPTIMALITY of SOLUTIONS

NOTE 1: We will show later that a stable marriage always exists.

NOTE 2: A stable marriage assignment does not need to be optimal for all.

EXAMPLE: Let us have three men M_1, M_2 and M_3 and three women W_1, W_2 and W_3 with preferences:

$$M_1 : W_2 W_1 W_3, \quad M_2 : W_3 W_2 W_1, \quad M_3 : W_1 W_3 W_2$$

$$W_1 : M_2 M_1 M_3, \quad W_2 : M_3 M_2 M_1, \quad W_3 : M_1 M_3 M_2$$

There are three stable solutions:

EXISTENCE and OPTIMALITY of SOLUTIONS

NOTE 1: We will show later that a stable marriage always exists.

NOTE 2: A stable marriage assignment does not need to be optimal for all.

EXAMPLE: Let us have three men M_1, M_2 and M_3 and three women W_1, W_2 and W_3 with preferences:

$$M_1 : W_2 W_1 W_3, \quad M_2 : W_3 W_2 W_1, \quad M_3 : W_1 W_3 W_2$$

$$W_1 : M_2 M_1 M_3, \quad W_2 : M_3 M_2 M_1, \quad W_3 : M_1 M_3 M_2$$

There are three stable solutions:

- All men get their first choice and all women their third one:

$$M_1 W_2, M_2 W_3, M_3 W_1$$

EXISTENCE and OPTIMALITY of SOLUTIONS

NOTE 1: We will show later that a stable marriage always exists.

NOTE 2: A stable marriage assignment does not need to be optimal for all.

EXAMPLE: Let us have three men M_1, M_2 and M_3 and three women W_1, W_2 and W_3 with preferences:

$$M_1 : W_2 W_1 W_3, \quad M_2 : W_3 W_2 W_1, \quad M_3 : W_1 W_3 W_2$$

$$W_1 : M_2 M_1 M_3, \quad W_2 : M_3 M_2 M_1, \quad W_3 : M_1 M_3 M_2$$

There are three stable solutions:

- All men get their first choice and all women their third one:

$$M_1 W_2, M_2 W_3, M_3 W_1$$

- All get their second choice:

EXISTENCE and OPTIMALITY of SOLUTIONS

NOTE 1: We will show later that a stable marriage always exists.

NOTE 2: A stable marriage assignment does not need to be optimal for all.

EXAMPLE: Let us have three men M_1, M_2 and M_3 and three women W_1, W_2 and W_3 with preferences:

$$M_1 : W_2 W_1 W_3, \quad M_2 : W_3 W_2 W_1, \quad M_3 : W_1 W_3 W_2$$

$$W_1 : M_2 M_1 M_3, \quad W_2 : M_3 M_2 M_1, \quad W_3 : M_1 M_3 M_2$$

There are three stable solutions:

- All men get their first choice and all women their third one:

$$M_1 W_2, M_2 W_3, M_3 W_1$$

- All get their second choice:

$$M_1 W_1, M_2 W_2, M_3 W_3$$

EXISTENCE and OPTIMALITY of SOLUTIONS

NOTE 1: We will show later that a stable marriage always exists.

NOTE 2: A stable marriage assignment does not need to be optimal for all.

EXAMPLE: Let us have three men M_1, M_2 and M_3 and three women W_1, W_2 and W_3 with preferences:

$$M_1 : W_2 W_1 W_3, \quad M_2 : W_3 W_2 W_1, \quad M_3 : W_1 W_3 W_2$$

$$W_1 : M_2 M_1 M_3, \quad W_2 : M_3 M_2 M_1, \quad W_3 : M_1 M_3 M_2$$

There are three stable solutions:

- All men get their first choice and all women their third one:

$$M_1 W_2, M_2 W_3, M_3 W_1$$

- All get their second choice:

$$M_1 W_1, M_2 W_2, M_3 W_3$$

- Women get their first choice and men the third one:

$$M_1 W_3, M_2 W_1, M_3 W_2$$

A naive, but not good enough, randomized algorithm

A naive, but not good enough, randomized algorithm

1. Start with some marriage of all.

A naive, but not good enough, randomized algorithm

1. Start with some marriage of all.
2. **until** marriage is stable **do** randomly choose a dissatisfied pair, marry them and also their partners together

A naive, but not good enough, randomized algorithm

1. Start with some marriage of all.
2. **until** marriage is stable **do** randomly choose a dissatisfied pair, marry them and also their partners together

Algorithm is not good because a loop can occur.

EXAMPLE 1

EXAMPLE 1

Let us have the followig preferences:

$$M_1 : W_3 W_2 W_4 W_1 \qquad M_2 : W_2 W_1 W_3 W_4$$
$$M_3 : W_2 W_4 W_1 W_3 \qquad M_4 : W_3 W_1 W_4 W_2$$

and

$$W_1 : M_1 M_2 M_4 M_3 \qquad W_2 : M_3 M_1 M_4 M_2$$
$$W_3 : M_3 M_2 M_4 M_1 \qquad W_4 : M_2 M_1 M_3 M_4$$

EXAMPLE 1

Let us have the followig preferences:

$$M_1 : W_3 W_2 W_4 W_1 \qquad M_2 : W_2 W_1 W_3 W_4$$
$$M_3 : W_2 W_4 W_1 W_3 \qquad M_4 : W_3 W_1 W_4 W_2$$

and

$$W_1 : M_1 M_2 M_4 M_3 \qquad W_2 : M_3 M_1 M_4 M_2$$
$$W_3 : M_3 M_2 M_4 M_1 \qquad W_4 : M_2 M_1 M_3 M_4$$

Successful developments of marriages:

$$\mathbf{M_1} W_1 \quad M_2 \mathbf{W_2} \quad M_3 W_3 \quad M_4 W_4 - -\text{unstable}$$

EXAMPLE 1

Let us have the followig preferences:

$$M_1 : W_3 W_2 W_4 W_1 \quad M_2 : W_2 W_1 W_3 W_4$$
$$M_3 : W_2 W_4 W_1 W_3 \quad M_4 : W_3 W_1 W_4 W_2$$

and

$$W_1 : M_1 M_2 M_4 M_3 \quad W_2 : M_3 M_1 M_4 M_2$$
$$W_3 : M_3 M_2 M_4 M_1 \quad W_4 : M_2 M_1 M_3 M_4$$

Successful developments of marriages:

$$\mathbf{M_1} W_1 \quad M_2 \mathbf{W_2} \quad M_3 W_3 \quad M_4 W_4 - -\text{unstable}$$

$$M_1 \mathbf{W_2} \quad M_2 W_1 \quad \mathbf{M_3} W_3 \quad M_4 W_4 - -\text{unstable}$$

EXAMPLE 1

Let us have the followig preferences:

$$M_1 : W_3 W_2 W_4 W_1 \qquad M_2 : W_2 W_1 W_3 W_4$$
$$M_3 : W_2 W_4 W_1 W_3 \qquad M_4 : W_3 W_1 W_4 W_2$$

and

$$W_1 : M_1 M_2 M_4 M_3 \qquad W_2 : M_3 M_1 M_4 M_2$$
$$W_3 : M_3 M_2 M_4 M_1 \qquad W_4 : M_2 M_1 M_3 M_4$$

Successful developments of marriages:

$$\mathbf{M_1} W_1 \quad M_2 \mathbf{W_2} \quad M_3 W_3 \quad M_4 W_4 \text{ --unstable}$$

$$M_1 \mathbf{W_2} \quad M_2 W_1 \quad \mathbf{M_3} W_3 \quad M_4 W_4 \text{ --unstable}$$

$$M_1 \mathbf{W_3} \quad M_2 W_1 \quad M_3 W_2 \quad \mathbf{M_4} W_4 \text{ --unstable}$$

EXAMPLE 1

Let us have the followig preferences:

$$M_1 : W_3 W_2 W_4 W_1 \quad M_2 : W_2 W_1 W_3 W_4$$
$$M_3 : W_2 W_4 W_1 W_3 \quad M_4 : W_3 W_1 W_4 W_2$$

and

$$W_1 : M_1 M_2 M_4 M_3 \quad W_2 : M_3 M_1 M_4 M_2$$
$$W_3 : M_3 M_2 M_4 M_1 \quad W_4 : M_2 M_1 M_3 M_4$$

Successful developments of marriages:

$$\mathbf{M_1} W_1 \quad M_2 \mathbf{W_2} \quad M_3 W_3 \quad M_4 W_4 --\text{unstable}$$

$$M_1 \mathbf{W_2} \quad M_2 W_1 \quad \mathbf{M_3} W_3 \quad M_4 W_4 --\text{unstable}$$

$$M_1 \mathbf{W_3} \quad M_2 W_1 \quad M_3 W_2 \quad \mathbf{M_4} W_4 --\text{unstable}$$

$$M_1 W_4 \quad M_2 W_1 \quad M_3 W_2 \quad M_4 W_3 --\text{!stable!}$$

EXAMPLE 2

For choices:

$$M_1 : W_2 W_1 W_3 \quad M_2 : \text{arbitrary} \quad M_3 : W_1 W_2 W_3$$

and

$$W_1 : M_1 M_3 M_2 \quad W_2 : M_3 M_1 M_2 \quad W_3 : \text{arbitrary}$$

we have the following cyclic development of marriages

EXAMPLE 2

For choices:

$$M_1 : W_2 W_1 W_3 \quad M_2 : \text{arbitrary} \quad M_3 : W_1 W_2 W_3$$

and

$$W_1 : M_1 M_3 M_2 \quad W_2 : M_3 M_1 M_2 \quad W_3 : \text{arbitrary}$$

we have the following cyclic development of marriages

$$\mathbf{M_1} W_1 \quad M_2 \mathbf{W_2} \quad M_3 W_3$$

EXAMPLE 2

For choices:

$$M_1 : W_2 W_1 W_3 \quad M_2 : \text{arbitrary} \quad M_3 : W_1 W_2 W_3$$

and

$$W_1 : M_1 M_3 M_2 \quad W_2 : M_3 M_1 M_2 \quad W_3 : \text{arbitrary}$$

we have the following cyclic development of marriages

$$\mathbf{M_1} W_1 \quad M_2 \mathbf{W_2} \quad M_3 W_3$$

$$M_1 \mathbf{W_2} \quad M_2 W_1 \quad \mathbf{M_3} W_3$$

EXAMPLE 2

For choices:

$$M_1 : W_2 W_1 W_3 \quad M_2 : \text{arbitrary} \quad M_3 : W_1 W_2 W_3$$

and

$$W_1 : M_1 M_3 M_2 \quad W_2 : M_3 M_1 M_2 \quad W_3 : \text{arbitrary}$$

we have the following cyclic development of marriages

$$\mathbf{M_1} W_1 \quad M_2 \mathbf{W_2} \quad M_3 W_3$$

$$M_1 \mathbf{W_2} \quad M_2 W_1 \quad \mathbf{M_3} W_3$$

$$M_1 W_3 \quad M_2 \mathbf{W_1} \quad \mathbf{M_3} W_2$$

EXAMPLE 2

For choices:

$$M_1 : W_2 W_1 W_3 \quad M_2 : \text{arbitrary} \quad M_3 : W_1 W_2 W_3$$

and

$$W_1 : M_1 M_3 M_2 \quad W_2 : M_3 M_1 M_2 \quad W_3 : \text{arbitrary}$$

we have the following cyclic development of marriages

$$\mathbf{M_1} W_1 \quad M_2 \mathbf{W_2} \quad M_3 W_3$$

$$M_1 \mathbf{W_2} \quad M_2 W_1 \quad \mathbf{M_3} W_3$$

$$M_1 W_3 \quad M_2 \mathbf{W_1} \quad \mathbf{M_3} W_2$$

$$\mathbf{M_1} W_3 \quad M_2 W_2 \quad M_3 \mathbf{W_1}$$

EXAMPLE 2

For choices:

$$M_1 : W_2 W_1 W_3 \quad M_2 : \text{arbitrary} \quad M_3 : W_1 W_2 W_3$$

and

$$W_1 : M_1 M_3 M_2 \quad W_2 : M_3 M_1 M_2 \quad W_3 : \text{arbitrary}$$

we have the following cyclic development of marriages

$$\mathbf{M_1} W_1 \quad M_2 \mathbf{W_2} \quad M_3 W_3$$

$$M_1 \mathbf{W_2} \quad M_2 W_1 \quad \mathbf{M_3} W_3$$

$$M_1 W_3 \quad M_2 \mathbf{W_1} \quad \mathbf{M_3} W_2$$

$$\mathbf{M_1} W_3 \quad M_2 W_2 \quad M_3 \mathbf{W_1}$$

$$M_1 W_1 \quad M_2 W_2 \quad M_3 W_3$$

EXAMPLE 3

For choices

$$M_1 : W_1 W_2 W_3 W_4 W_5 \quad M_2 : W_2 W_3 W_4 W_5 W_1 \quad M_3 : W_3 W_4 W_5 W_1 W_2$$
$$M_4 : W_4 W_5 W_1 W_2 W_3 \quad M_5 : W_5 W_1 W_2 W_3 W_4 W_5$$

and

$$W_1 : M_2 M_3 M_4 M_5 M_1 \quad W_2 : M_3 M_4 M_5 M_1 M_2 \quad W_3 : M_4 M_5 M_1 M_2 M_3$$
$$W_4 : M_5 M_1 M_2 M_3 M_4 \quad W_5 : M_1 M_2 M_3 M_4 M_5$$

we have

EXAMPLE 3

For choices

$$M_1 : W_1 W_2 W_3 W_4 W_5 \quad M_2 : W_2 W_3 W_4 W_5 W_1 \quad M_3 : W_3 W_4 W_5 W_1 W_2$$
$$M_4 : W_4 W_5 W_1 W_2 W_3 \quad M_5 : W_5 W_1 W_2 W_3 W_4 W_5$$

and

$$W_1 : M_2 M_3 M_4 M_5 M_1 \quad W_2 : M_3 M_4 M_5 M_1 M_2 \quad W_3 : M_4 M_5 M_1 M_2 M_3$$
$$W_4 : M_5 M_1 M_2 M_3 M_4 \quad W_5 : M_1 M_2 M_3 M_4 M_5$$

we have exactly 5 stable marriages

$$\begin{array}{ccccc}
M_1 W_1 & M_2 W_2 & M_3 W_3 & M_4 W_4 & M_5 W_5 \\
M_1 W_2 & M_2 W_3 & M_3 W_4 & M_4 W_5 & M_5 W_1 \\
M_1 W_3 & M_2 W_4 & M_3 W_5 & M_4 W_1 & M_5 W_2 \\
M_1 W_4 & M_2 W_5 & M_4 W_1 & M_4 W_2 & M_5 W_3 \\
M_1 W_5 & M_2 W_1 & M_4 W_2 & M_5 W_3 & M_5 W_4
\end{array}$$

EXAMPLE 3

For choices

$$M_1 : W_1 W_2 W_3 W_4 W_5 \quad M_2 : W_2 W_3 W_4 W_5 W_1 \quad M_3 : W_3 W_4 W_5 W_1 W_2$$
$$M_4 : W_4 W_5 W_1 W_2 W_3 \quad M_5 : W_5 W_1 W_2 W_3 W_4 W_5$$

and

$$W_1 : M_2 M_3 M_4 M_5 M_1 \quad W_2 : M_3 M_4 M_5 M_1 M_2 \quad W_3 : M_4 M_5 M_1 M_2 M_3$$
$$W_4 : M_5 M_1 M_2 M_3 M_4 \quad W_5 : M_1 M_2 M_3 M_4 M_5$$

we have exactly 5 stable marriages

$$\begin{array}{ccccc} M_1 W_1 & M_2 W_2 & M_3 W_3 & M_4 W_4 & M_5 W_5 \\ M_1 W_2 & M_2 W_3 & M_3 W_4 & M_4 W_5 & M_5 W_1 \\ M_1 W_3 & M_2 W_4 & M_3 W_5 & M_4 W_1 & M_5 W_2 \\ M_1 W_4 & M_2 W_5 & M_4 W_1 & M_4 W_2 & M_5 W_3 \\ M_1 W_5 & M_2 W_1 & M_4 W_2 & M_5 W_3 & M_5 W_4 \end{array}$$

In the x-th of the above marriages each man is married with his x-th choice.

PROPOSAL ALGORITHMS

PROPOSAL ALGORITHMS

The naive approach – to start with an arbitrary marriage and to try to stabilize it by pairing up dissatisfied couples – does not always work.

PROPOSAL ALGORITHMS

The naive approach – to start with an arbitrary marriage and to try to stabilize it by pairing up dissatisfied couples – does not always work.

MEN PROPOSAL ALGORITHM - **"man proposes, woman disposes"**

PROPOSAL ALGORITHMS

The naive approach – to start with an arbitrary marriage and to try to stabilize it by pairing up dissatisfied couples – does not always work.

MEN PROPOSAL ALGORITHM - **"man proposes, woman disposes"**

Assume that all men are numbered somehow.

PROPOSAL ALGORITHMS

The naive approach – to start with an arbitrary marriage and to try to stabilize it by pairing up dissatisfied couples – does not always work.

MEN PROPOSAL ALGORITHM - **"man proposes, woman disposes"**
Assume that all men are numbered somehow.

At any step of the algorithm (due to Gale-Shapley), there will be a partial marriage, and the lowest-number unmarried man M proposes "marriage" to the most desirable women W on his list who has not rejected him yet.

PROPOSAL ALGORITHMS

The naive approach – to start with an arbitrary marriage and to try to stabilize it by pairing up dissatisfied couples – does not always work.

MEN PROPOSAL ALGORITHM - **"man proposes, woman disposes"**
Assume that all men are numbered somehow.

At any step of the algorithm (due to Gale-Shapley), there will be a partial marriage, and the lowest-number unmarried man M proposes "marriage" to the most desirable women W on his list who has not rejected him yet. The woman W then decides whether to accept his proposal or to reject it.

PROPOSAL ALGORITHMS

The naive approach – to start with an arbitrary marriage and to try to stabilize it by pairing up dissatisfied couples – does not always work.

MEN PROPOSAL ALGORITHM - **"man proposes, woman disposes"**
Assume that all men are numbered somehow.

At any step of the algorithm (due to Gale-Shapley), there will be a partial marriage, and the lowest-number unmarried man M proposes "marriage" to the most desirable women W on his list who has not rejected him yet. The woman W then decides whether to accept his proposal or to reject it.

The women W accepts the proposal if

PROPOSAL ALGORITHMS

The naive approach – to start with an arbitrary marriage and to try to stabilize it by pairing up dissatisfied couples – does not always work.

MEN PROPOSAL ALGORITHM - **"man proposes, woman disposes"**
Assume that all men are numbered somehow.

At any step of the algorithm (due to Gale-Shapley), there will be a partial marriage, and the lowest-number unmarried man M proposes "marriage" to the most desirable women W on his list who has not rejected him yet. The woman W then decides whether to accept his proposal or to reject it.

The women W accepts the proposal if

- she is not yet married or

PROPOSAL ALGORITHMS

The naive approach – to start with an arbitrary marriage and to try to stabilize it by pairing up dissatisfied couples – does not always work.

MEN PROPOSAL ALGORITHM - **"man proposes, woman disposes"**
Assume that all men are numbered somehow.

At any step of the algorithm (due to Gale-Shapley), there will be a partial marriage, and the lowest-number unmarried man M proposes "marriage" to the most desirable women W on his list who has not rejected him yet. The woman W then decides whether to accept his proposal or to reject it.

The women W accepts the proposal if

- she is not yet married or
- she likes M more than her current partner.

The algorithm repeats the process and terminates after every person has been married.

PROPOSAL ALGORITHMS

The naive approach – to start with an arbitrary marriage and to try to stabilize it by pairing up dissatisfied couples – does not always work.

MEN PROPOSAL ALGORITHM - **"man proposes, woman disposes"**

Assume that all men are numbered somehow.

At any step of the algorithm (due to Gale-Shapley), there will be a partial marriage, and the lowest-number unmarried man M proposes "marriage" to the most desirable women W on his list who has not rejected him yet. The woman W then decides whether to accept his proposal or to reject it.

The women W accepts the proposal if

- she is not yet married or
- she likes M more than her current partner.

The algorithm repeats the process and terminates after every person has been married. It is a linear time algorithm, concerning the worst case complexity.

It is easy to see that the process terminates and resulting marriage is stable.

CORRECTNESS of the ALGORITHM

CORRECTNESS of the ALGORITHM

Everyone gets married Observe that once a women gets married she will stay married (though she can change her partners - even several times).

CORRECTNESS of the ALGORITHM

Everyone gets married Observe that once a women gets married she will stay married (though she can change her partners - even several times).

It cannot be the case that at the end there is a man and a woman who are not married. Indeed, the men would have proposed her marriage at some point and being unmarried she could not refused him.

CORRECTNESS of the ALGORITHM

Everyone gets married Observe that once a women gets married she will stay married (though she can change her partners - even several times).

It cannot be the case that at the end there is a man and a woman who are not married. Indeed, the men would have proposed her marriage at some point and being unmarried she could not refused him.

Final marriage is stable Indeed, let at the end M be a men and W a women who are married, but not to each other and they are dissatisfied.

CORRECTNESS of the ALGORITHM

Everyone gets married Observe that once a women gets married she will stay married (though she can change her partners - even several times).

It cannot be the case that at the end there is a man and a woman who are not married. Indeed, the men would have proposed her marriage at some point and being unmarried she could not refused him.

Final marriage is stable Indeed, let at the end M be a men and W a women who are married, but not to each other and they are dissatisfied. If M prefers W over his current partner, he must have proposed marriage to W before he did that to his current partner.

CORRECTNESS of the ALGORITHM

Everyone gets married Observe that once a women gets married she will stay married (though she can change her partners - even several times).

It cannot be the case that at the end there is a man and a woman who are not married. Indeed, the men would have proposed her marriage at some point and being unmarried she could not refused him.

Final marriage is stable Indeed, let at the end M be a men and W a women who are married, but not to each other and they are dissatisfied. If M prefers W over his current partner, he must have proposed marriage to W before he did that to his current partner. If W accepted his proposal yet is not married with him at the end, she must have changed him for someone she likes more and therefore she cannot like M more than her current partner.

CORRECTNESS of the ALGORITHM

Everyone gets married Observe that once a women gets married she will stay married (though she can change her partners - even several times).

It cannot be the case that at the end there is a man and a woman who are not married. Indeed, the men would have proposed her marriage at some point and being unmarried she could not refused him.

Final marriage is stable Indeed, let at the end M be a men and W a women who are married, but not to each other and they are dissatisfied. If M prefers W over his current partner, he must have proposed marriage to W before he did that to his current partner. If W accepted his proposal yet is not married with him at the end, she must have changed him for someone she likes more and therefore she cannot like M more than her current partner. If W rejected his proposal, she was already married with someone she liked more than M.

COMPLEXITY of the PROPOSAL ALGORITHM

COMPLEXITY of the PROPOSAL ALGORITHM

At each proposal step one women is eliminated from a man list. Total number of proposals is therefore at most n^2.

COMPLEXITY of the PROPOSAL ALGORITHM

At each proposal step one women is eliminated from a man list. Total number of proposals is therefore at most n^2.

The result of the men-proposal algorithm does not depend on the order men are chosen to make their proposals.

PROPERTIES of the GALE-SHAPLEY-ALGORITHM

PROPERTIES of the GALE-SHAPLEY-ALGORITHM

Gale-Shapley marriage is men-optimal and women-pessimal. To see that consider the following definition of a feasible marriage.

PROPERTIES of the GALE-SHAPLEY-ALGORITHM

Gale-Shapley marriage is men-optimal and women-pessimal. To see that consider the following definition of a feasible marriage.

A marriage between a man A and a woman B is called feasible if there exists a stable pairing (marriage) in which A and B are married.

PROPERTIES of the GALE-SHAPLEY-ALGORITHM

Gale-Shapley marriage is men-optimal and women-pessimal. To see that consider the following definition of a feasible marriage.

A marriage between a man A and a woman B is called feasible if there exists a stable pairing (marriage) in which A and B are married.

It is said that a marriage is men-optimal if every man is married with his highest ranked feasible partner.

PROPERTIES of the GALE-SHAPLEY-ALGORITHM

Gale-Shapley marriage is men-optimal and women-pessimal. To see that consider the following definition of a feasible marriage.

A marriage between a man A and a woman B is called feasible if there exists a stable pairing (marriage) in which A and B are married.

It is said that a marriage is men-optimal if every man is married with his highest ranked feasible partner.

It is said that a marriage is women-pessimal if each woman is married with her lowest ranked feasible partner.

SOME APPLICATIONS - I.

SOME APPLICATIONS - I.

National residency matching program.

This program places applicants for postgraduate medical training positions into residency programs at teaching hospitals throughout US.

SOME APPLICATIONS - II.

SOME APPLICATIONS - II.

- Dental residencies and medical specialities in the USA, Canada and parts of UK

SOME APPLICATIONS - II.

- Dental residencies and medical specialities in the USA, Canada and parts of UK
- National university entrance exam in Iran

SOME APPLICATIONS - II.

- Dental residencies and medical specialities in the USA, Canada and parts of UK
- National university entrance exam in Iran
- Placement of Canadian lawyers in Ontario and Alberta

SOME APPLICATIONS - II.

- Dental residencies and medical specialities in the USA, Canada and parts of UK
- National university entrance exam in Iran
- Placement of Canadian lawyers in Ontario and Alberta
- Matching of new reform rabbis to their first congregation

SOME APPLICATIONS - II.

- Dental residencies and medical specialities in the USA, Canada and parts of UK
- National university entrance exam in Iran
- Placement of Canadian lawyers in Ontario and Alberta
- Matching of new reform rabbis to their first congregation
- Assignment of students to high-schools in NYC

STABLE HUSBANDS

STABLE HUSBANDS

A stable husband of a woman, with respect to a given rankings, is a man she can be married with in a stable marriage.

STABLE HUSBANDS

A stable husband of a woman, with respect to a given rankings, is a man she can be married with in a stable marriage.

D. E. Knuth and et al. showed that

STABLE HUSBANDS

A stable husband of a woman, with respect to a given rankings, is a man she can be married with in a stable marriage.

D. E. Knuth and et al. showed that

In case of n men and n women, any woman has at least $(\frac{1}{2} - \epsilon) \ln n$ and at most $(1 + \epsilon) \ln n$ different stable husbands in the set of all Gale-Shapley stable matchings defined by these rankings, with probability approaching 1 as $n \to \infty$, if ϵ is any positive constant.

STABLE HUSBANDS

A stable husband of a woman, with respect to a given rankings, is a man she can be married with in a stable marriage.

D. E. Knuth and et al. showed that

In case of n men and n women, any woman has at least $(\frac{1}{2} - \epsilon) \ln n$ and at most $(1 + \epsilon) \ln n$ different stable husbands in the set of all Gale-Shapley stable matchings defined by these rankings, with probability approaching 1 as $n \to \infty$, if ϵ is any positive constant.

There is an algorithm that outputs all stable husbands of a given women.

RANDOMIZED VERSIONS of the PROPOSAL ALGORITHM

RANDOMIZED VERSIONS of the PROPOSAL ALGORITHM

Next goal: The average-case analysis of the proposal algorithm under the assumptions:

RANDOMIZED VERSIONS of the PROPOSAL ALGORITHM

Next goal: The average-case analysis of the proposal algorithm under the assumptions:

men's lists are chosen independently and randomly,

RANDOMIZED VERSIONS of the PROPOSAL ALGORITHM

Next goal: The average-case analysis of the proposal algorithm under the assumptions:

men's lists are chosen independently and randomly,
women's lists can be arbitrary, but are fixed in advance.

RANDOMIZED VERSIONS of the PROPOSAL ALGORITHM

Next goal: The average-case analysis of the proposal algorithm under the assumptions:

men's lists are chosen independently and randomly,
women's lists can be arbitrary, but are fixed in advance.

Let T_p be the random variable that denote the number of proposals made during the execution of the *Proposal algorithm* – what is proportional to the overall time of algorithm.

RANDOMIZED VERSIONS of the PROPOSAL ALGORITHM

Next goal: The average-case analysis of the proposal algorithm under the assumptions:

men's lists are chosen independently and randomly,
women's lists can be arbitrary, but are fixed in advance.

Let $\mathrm{T_p}$ be the random variable that denote the number of proposals made during the execution of the *Proposal algorithm* – what is proportional to the overall time of algorithm.

Distribution of $\mathrm{T_p}$ seems to be very difficult to determine or even to analyse.

Our goal is to show that the expected value of the number of proposals is about $\mathcal{O}(n \lg n)$.

Game “CLOCK SOLITAIRE”

We illustrate first, on a simple card game, a simple technique that allows to analyse randomized algorithms with seemingly complex behaviour.

Game “CLOCK SOLITAIRE”

We illustrate first, on a simple card game, a simple technique that allows to analyse randomized algorithms with seemingly complex behaviour.

1. Game ”Clock Solitaire”

Game "CLOCK SOLITAIRE"

We illustrate first, on a simple card game, a simple technique that allows to analyse randomized algorithms with seemingly complex behaviour.

1. Game "Clock Solitaire"

A standard deck of 52 cards is randomly shuffled and then divided into 13 piles (columns) of 4 cards each. Each pile is arbitrarily labeled with a distinct symbol from $\{A, 2, \ldots, 10, J, Q, K\}$

Game "CLOCK SOLITAIRE"

We illustrate first, on a simple card game, a simple technique that allows to analyse randomized algorithms with seemingly complex behaviour.

1. Game "Clock Solitaire"

A standard deck of 52 cards is randomly shuffled and then divided into 13 piles (columns) of 4 cards each. Each pile is arbitrarily labeled with a distinct symbol from $\{A, 2, \ldots, 10, J, Q, K\}$

A	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**J**	**Q**	**K**
A	2	3	4	5	6	7	8	9	10	J	Q	K
4	3	4	5	6	7	8	9	10	J	Q	K	A
2	4	5	6	7	8	9	10	J	Q	K	A	3
3	5	6	7	8	9	10	J	Q	K	A	2	2

Game "CLOCK SOLITAIRE"

We illustrate first, on a simple card game, a simple technique that allows to analyse randomized algorithms with seemingly complex behaviour.

1. Game "Clock Solitaire"

A standard deck of 52 cards is randomly shuffled and then divided into 13 piles (columns) of 4 cards each. Each pile is arbitrarily labeled with a distinct symbol from $\{A, 2, \ldots, 10, J, Q, K\}$

A	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**J**	**Q**	**K**
A	2	3	4	5	6	7	8	9	10	J	Q	K
4	3	4	5	6	7	8	9	10	J	Q	K	A
2	4	5	6	7	8	9	10	J	Q	K	A	3
3	5	6	7	8	9	10	J	Q	K	A	2	2

Playing the game:

Game "CLOCK SOLITAIRE"

We illustrate first, on a simple card game, a simple technique that allows to analyse randomized algorithms with seemingly complex behaviour.

1. Game "Clock Solitaire"

A standard deck of 52 cards is randomly shuffled and then divided into 13 piles (columns) of 4 cards each. Each pile is arbitrarily labeled with a distinct symbol from $\{A, 2, \ldots, 10, J, Q, K\}$

A	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**J**	**Q**	**K**
A	2	3	4	5	6	7	8	9	10	J	Q	K
4	3	4	5	6	7	8	9	10	J	Q	K	A
2	4	5	6	7	8	9	10	J	Q	K	A	3
3	5	6	7	8	9	10	J	Q	K	A	2	2

Playing the game: On the first move a card is drawn from the pile labeled K.

Game “CLOCK SOLITAIRE”

We illustrate first, on a simple card game, a simple technique that allows to analyse randomized algorithms with seemingly complex behaviour.

1. Game ”Clock Solitaire”

A standard deck of 52 cards is randomly shuffled and then divided into 13 piles (columns) of 4 cards each. Each pile is arbitrarily labeled with a distinct symbol from $\{\mathrm{A}, 2, \ldots, 10, \mathrm{J}, \mathrm{Q}, \mathrm{K}\}$

A	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**J**	**Q**	**K**
A	2	3	4	5	6	7	8	9	10	J	Q	K
4	3	4	5	6	7	8	9	10	J	Q	K	A
2	4	5	6	7	8	9	10	J	Q	K	A	3
3	5	6	7	8	9	10	J	Q	K	A	2	2

Playing the game: On the first move a card is drawn from the pile labeled K.

At each subsequent move, a card is drawn, by the player, from the pile whose label is the face value of the card at the previous move.

Game “CLOCK SOLITAIRE”

We illustrate first, on a simple card game, a simple technique that allows to analyse randomized algorithms with seemingly complex behaviour.

1. Game "Clock Solitaire"

A standard deck of 52 cards is randomly shuffled and then divided into 13 piles (columns) of 4 cards each. Each pile is arbitrarily labeled with a distinct symbol from $\{\mathrm{A}, 2, \ldots, 10, \mathrm{J}, \mathrm{Q}, \mathrm{K}\}$

A	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**J**	**Q**	**K**
A	2	3	4	5	6	7	8	9	10	J	Q	K
4	3	4	5	6	7	8	9	10	J	Q	K	A
2	4	5	6	7	8	9	10	J	Q	K	A	3
3	5	6	7	8	9	10	J	Q	K	A	2	2

Playing the game: On the first move a card is drawn from the pile labeled K.

At each subsequent move, a card is drawn, by the player, from the pile whose label is the face value of the card at the previous move.

The game ends, if the player makes an attempt to draw a card from an empty pile.

Game “CLOCK SOLITAIRE”

We illustrate first, on a simple card game, a simple technique that allows to analyse randomized algorithms with seemingly complex behaviour.

1. Game "Clock Solitaire"

A standard deck of 52 cards is randomly shuffled and then divided into 13 piles (columns) of 4 cards each. Each pile is arbitrarily labeled with a distinct symbol from $\{A, 2, \ldots, 10, J, Q, K\}$

A	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**J**	**Q**	**K**
A	2	3	4	5	6	7	8	9	10	J	Q	K
4	3	4	5	6	7	8	9	10	J	Q	K	A
2	4	5	6	7	8	9	10	J	Q	K	A	3
3	5	6	7	8	9	10	J	Q	K	A	2	2

Playing the game: On the first move a card is drawn from the pile labeled K.

At each subsequent move, a card is drawn, by the player, from the pile whose label is the face value of the card at the previous move.

The game ends, if the player makes an attempt to draw a card from an empty pile.

Player wins the game if, on termination, all 52 cards have been drawn.

Game "CLOCK SOLITAIRE"

We illustrate first, on a simple card game, a simple technique that allows to analyse randomized algorithms with seemingly complex behaviour.

1. Game "Clock Solitaire"

A standard deck of 52 cards is randomly shuffled and then divided into 13 piles (columns) of 4 cards each. Each pile is arbitrarily labeled with a distinct symbol from $\{A, 2, \ldots, 10, J, Q, K\}$

A	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**J**	**Q**	**K**
A	2	3	4	5	6	7	8	9	10	J	Q	K
4	3	4	5	6	7	8	9	10	J	Q	K	A
2	4	5	6	7	8	9	10	J	Q	K	A	3
3	5	6	7	8	9	10	J	Q	K	A	2	2

Playing the game: On the first move a card is drawn from the pile labeled K.

At each subsequent move, a card is drawn, by the player, from the pile whose label is the face value of the card at the previous move.

The game ends, if the player makes an attempt to draw a card from an empty pile.

Player wins the game if, on termination, all 52 cards have been drawn. In all other cases the player looses the game.

Game "CLOCK SOLITAIRE"

We illustrate first, on a simple card game, a simple technique that allows to analyse randomized algorithms with seemingly complex behaviour.

1. Game "Clock Solitaire"

A standard deck of 52 cards is randomly shuffled and then divided into 13 piles (columns) of 4 cards each. Each pile is arbitrarily labeled with a distinct symbol from $\{\mathrm{A}, 2, \ldots, 10, \mathrm{J}, \mathrm{Q}, \mathrm{K}\}$

A	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**J**	**Q**	**K**
A	2	3	4	5	6	7	8	9	10	J	Q	K
4	3	4	5	6	7	8	9	10	J	Q	K	A
2	4	5	6	7	8	9	10	J	Q	K	A	3
3	5	6	7	8	9	10	J	Q	K	A	2	2

Playing the game: On the first move a card is drawn from the pile labeled K.

At each subsequent move, a card is drawn, by the player, from the pile whose label is the face value of the card at the previous move.

The game ends, if the player makes an attempt to draw a card from an empty pile.

Player wins the game if, on termination, all 52 cards have been drawn. In all other cases the player looses the game. What is the probability to win the game?

A	2	3	4	5	6	7	8	9	10	J	Q	K
A	2	3	4	5	6	7	8	9	10	J	Q	K
2	3	4	5	6	7	8	9	10	J	Q	K	A
3	4	5	6	7	8	9	10	J	Q	K	A	2
4	5	6	7	8	9	10	J	Q	K	A	2	3

A	2	3	4	5	6	7	8	9	10	J	Q	K
A	2	3	4	5	6	7	8	9	10	J	Q	K
2	3	4	5	6	7	8	9	10	J	Q	K	A
3	4	5	6	7	8	9	10	J	Q	K	A	.
4	5	6	7	8	9	10	J	Q	K	A	2	3

A	2	3	4	5	6	7	8	9	10	J	Q	K
A	2	3	4	5	6	7	8	9	10	J	Q	K
2	.	4	5	6	7	8	9	10	J	Q	K	A
3	4	5	6	7	8	9	10	J	Q	K	A	.
4	5	6	7	8	9	10	J	Q	K	A	2	3

PRINCIPLE od "DEFERRED RANDOM DECISIONS"

1. **Game "Clock Solitaire" – repetition**

PRINCIPLE od "DEFERRED RANDOM DECISIONS"

1. Game "Clock Solitaire" – repetition

A standard deck of 52 cards is randomly shuffled and then divided into 13 piles of 4 cards each.

PRINCIPLE od "DEFERRED RANDOM DECISIONS"

1. Game "Clock Solitaire" – repetition

A standard deck of 52 cards is randomly shuffled and then divided into 13 piles of 4 cards each. Each pile is arbitrarily labeled with a distinct symbol from $\{A, 2, \ldots, 10, J, Q, K\}$

PRINCIPLE od "DEFERRED RANDOM DECISIONS"

1. Game "Clock Solitaire" – repetition

A standard deck of 52 cards is randomly shuffled and then divided into 13 piles of 4 cards each. Each pile is arbitrarily labeled with a distinct symbol from $\{A, 2, \ldots, 10, J, Q, K\}$

At each subsequent move, a card is drawn from the pile whose label is the face value of the card at the previous move.

PRINCIPLE od "DEFERRED RANDOM DECISIONS"

1. Game "Clock Solitaire" – repetition

A standard deck of 52 cards is randomly shuffled and then divided into 13 piles of 4 cards each. Each pile is arbitrarily labeled with a distinct symbol from $\{A, 2, \ldots, 10, J, Q, K\}$

At each subsequent move, a card is drawn from the pile whose label is the face value of the card at the previous move.

The game ends, if an attempt is made to draw a card from an empty pile.

PRINCIPLE od "DEFERRED RANDOM DECISIONS"

1. Game "Clock Solitaire" – repetition

A standard deck of 52 cards is randomly shuffled and then divided into 13 piles of 4 cards each. Each pile is arbitrarily labeled with a distinct symbol from $\{\mathrm{A}, 2, \ldots, 10, \mathrm{J}, \mathrm{Q}, \mathrm{K}\}$

At each subsequent move, a card is drawn from the pile whose label is the face value of the card at the previous move.

The game ends, if an attempt is made to draw a card from an empty pile.

Observe that our game always terminates in an attempt to draw a card from the K-pile. (Why?)

PRINCIPLE od "DEFERRED RANDOM DECISIONS"

1. Game "Clock Solitaire" – repetition

A standard deck of 52 cards is randomly shuffled and then divided into 13 piles of 4 cards each. Each pile is arbitrarily labeled with a distinct symbol from $\{$A, 2, $\ldots$, 10, J, Q, K$\}$

At each subsequent move, a card is drawn from the pile whose label is the face value of the card at the previous move.

The game ends, if an attempt is made to draw a card from an empty pile.

Observe that our game always terminates in an attempt to draw a card from the K-pile. (Why?)

ANALYSIS of ALGORITHM How to choose the probability space? Let the random choices unfold with progress of the game: that is at any step each of the yet unseen cards is likely to appear.

PRINCIPLE od "DEFERRED RANDOM DECISIONS"

1. Game "Clock Solitaire" – repetition

A standard deck of 52 cards is randomly shuffled and then divided into 13 piles of 4 cards each. Each pile is arbitrarily labeled with a distinct symbol from $\{A, 2, \ldots, 10, J, Q, K\}$

At each subsequent move, a card is drawn from the pile whose label is the face value of the card at the previous move.

The game ends, if an attempt is made to draw a card from an empty pile.

Observe that our game always terminates in an attempt to draw a card from the K-pile. (Why?)

ANALYSIS of ALGORITHM How to choose the probability space? Let the random choices unfold with progress of the game: that is at any step each of the yet unseen cards is likely to appear.

Thus, the process of playing this game is equivalent to the process of repeatedly drawing cards uniformly and randomly from the deck of 52 cards.

PRINCIPLE od "DEFERRED RANDOM DECISIONS"

1. Game "Clock Solitaire" – repetition

A standard deck of 52 cards is randomly shuffled and then divided into 13 piles of 4 cards each. Each pile is arbitrarily labeled with a distinct symbol from $\{A, 2, \ldots, 10, J, Q, K\}$

At each subsequent move, a card is drawn from the pile whose label is the face value of the card at the previous move.

The game ends, if an attempt is made to draw a card from an empty pile.

Observe that our game always terminates in an attempt to draw a card from the K-pile. (Why?)

ANALYSIS of ALGORITHM How to choose the probability space? Let the random choices unfold with progress of the game: that is at any step each of the yet unseen cards is likely to appear.

Thus, the process of playing this game is equivalent to the process of repeatedly drawing cards uniformly and randomly from the deck of 52 cards. A **winning game** corresponds to the situation where the first 51 cards drawn in this fashion contain exactly 3 kings!

PRINCIPLE od "DEFERRED RANDOM DECISIONS"

1. Game "Clock Solitaire" – repetition

A standard deck of 52 cards is randomly shuffled and then divided into 13 piles of 4 cards each. Each pile is arbitrarily labeled with a distinct symbol from $\{A, 2, \ldots, 10, J, Q, K\}$

At each subsequent move, a card is drawn from the pile whose label is the face value of the card at the previous move.

The game ends, if an attempt is made to draw a card from an empty pile.

Observe that our game always terminates in an attempt to draw a card from the K-pile. (Why?)

ANALYSIS of ALGORITHM How to choose the probability space? Let the random choices unfold with progress of the game: that is at any step each of the yet unseen cards is likely to appear.

Thus, the process of playing this game is equivalent to the process of repeatedly drawing cards uniformly and randomly from the deck of 52 cards. A **winning game** corresponds to the situation where the first 51 cards drawn in this fashion contain exactly 3 kings!

Probability of winning our game is therefore, clearly, 1/13.

Principle of Deferred Decisions

Principle of Deferred Decisions

At solving problems, it is not necessary that the entire set of random choices has to be made in advance,

Principle of Deferred Decisions

At solving problems, it is not necessary that the entire set of random choices has to be made in advance,

rather,

Principle of Deferred Decisions

At solving problems, it is not necessary that the entire set of random choices has to be made in advance,

rather,

it is sufficient that **at each step of the algorithm we fix only that random choice that needs to be revealed at that step**.

ANALYSIS of RANDOMIZED VERSION of PROPOSAL ALGORITHM 1/2

ANALYSIS of RANDOMIZED VERSION of PROPOSAL ALGORITHM 1/2

Principle of deferred decision: Do not assume that entire set of random choices is made in advance. Rather, at each step of the process fix only that random choices that must be revealed at that step to the algorithm.

ANALYSIS of RANDOMIZED VERSION of PROPOSAL ALGORITHM 1/2

Principle of deferred decision: Do not assume that entire set of random choices is made in advance. Rather, at each step of the process fix only that random choices that must be revealed at that step to the algorithm.

An application to the Proposal Algorithm: We will remove dependencies by do not assuming that men have chosen their preference lists in advance.

ANALYSIS of RANDOMIZED VERSION of PROPOSAL ALGORITHM 1/2

Principle of deferred decision: Do not assume that entire set of random choices is made in advance. Rather, at each step of the process fix only that random choices that must be revealed at that step to the algorithm.

An application to the Proposal Algorithm: We will remove dependencies by do not assuming that men have chosen their preference lists in advance.

We will assume that each time a man has to make a proposal he picks a random woman from the list od women not already proposed by him, and proceeds to propose her. (Clearly this is equivalent to choosing a random preference list prior the execution of the algorithm.)

ANALYSIS of RANDOMIZED VERSION of PROPOSAL ALGORITHM 1/2

Principle of deferred decision: Do not assume that entire set of random choices is made in advance. Rather, at each step of the process fix only that random choices that must be revealed at that step to the algorithm.

An application to the Proposal Algorithm: We will remove dependencies by do not assuming that men have chosen their preference lists in advance.

We will assume that each time a man has to make a proposal he picks a random woman from the list od women not already proposed by him, and proceeds to propose her. (Clearly this is equivalent to choosing a random preference list prior the execution of the algorithm.)

The only dependency that remains is that the random choice of a women at any step depends on the proposals made so far by the current proposer.

ANALYSIS of RANDOMIZED VERSION of PROPOSAL ALGORITHM 1/2

Principle of deferred decision: Do not assume that entire set of random choices is made in advance. Rather, at each step of the process fix only that random choices that must be revealed at that step to the algorithm.

An application to the Proposal Algorithm: We will remove dependencies by do not assuming that men have chosen their preference lists in advance.

We will assume that each time a man has to make a proposal he picks a random woman from the list od women not already proposed by him, and proceeds to propose her. (Clearly this is equivalent to choosing a random preference list prior the execution of the algorithm.)

The only dependency that remains is that the random choice of a women at any step depends on the proposals made so far by the current proposer.

To eliminate the above dependency let us change the algorithm. Each time a man makes proposal he chooses randomly a woman from the set of all women. Call this new algorithm **Amnesiac Algorithm**.

ANALYSIS of RANDOMIZED VERSION of PROPOSAL ALGORITHM 2/2

ANALYSIS of RANDOMIZED VERSION of PROPOSAL ALGORITHM 2/2

Let $\mathrm{T_A}(\mathrm{T_P})$ be the number of proposal made by the Amnesiac (Proposal) algorithm. It is obvious that for all m

$$Pr[\mathrm{T_A} > m] \geq Pr[\mathrm{T_P} \geq m]$$

and therefore an upper bound on T_A is an upper bound on T_P.

ANALYSIS of RANDOMIZED VERSION of PROPOSAL ALGORITHM 2/2

Let $\mathrm{T_A}(\mathrm{T_P})$ be the number of proposal made by the Amnesiac (Proposal) algorithm. It is obvious that for all m

$$Pr[\mathrm{T_A} > m] \geq Pr[\mathrm{T_P} \geq m]$$

and therefore an upper bound on T_A is an upper bound on T_P.

The advantage of analyzing $\mathrm{T_A}$ is that we need only to count the total number of proposals made – without regard to the name of the proposer at each stage.

ANALYSIS of RANDOMIZED VERSION of PROPOSAL ALGORITHM 2/2

Let $\mathrm{T_A}(\mathrm{T_P})$ be the number of proposal made by the Amnesiac (Proposal) algorithm. It is obvious that for all m

$$Pr[\mathrm{T_A} > m] \geq Pr[\mathrm{T_P} \geq m]$$

and therefore an upper bound on T_A is an upper bound on T_P.
The advantage of analyzing $\mathrm{T_A}$ is that we need only to count the total number of proposals made – without regard to the name of the proposer at each stage.

New algorithm terminates with a stable marriage once each woman has received at least one proposal (for a "marriage").

ANALYSIS of RANDOMIZED VERSION of PROPOSAL ALGORITHM 2/2

Let $\mathrm{T_A}(\mathrm{T_P})$ be the number of proposal made by the Amnesiac (Proposal) algorithm. It is obvious that for all m

$$Pr[\mathrm{T_A} > m] \geq Pr[\mathrm{T_P} \geq m]$$

and therefore an upper bound on T_A is an upper bound on T_P.
The advantage of analyzing $\mathrm{T_A}$ is that we need only to count the total number of proposals made – without regard to the name of the proposer at each stage.

New algorithm terminates with a stable marriage once each woman has received at least one proposal (for a "marriage").

The task to determine a good upper bound of T_A is a special case of the task to determine such a bound for so-called **Coupon Selection Problem** discussed next.

ANALYSIS of RANDOMIZED VERSION of PROPOSAL ALGORITHM 2/2

Let $\mathrm{T_A}(\mathrm{T_P})$ be the number of proposal made by the Amnesiac (Proposal) algorithm. It is obvious that for all m

$$Pr[\mathrm{T_A} > m] \geq Pr[\mathrm{T_P} \geq m]$$

and therefore an upper bound on T_A is an upper bound on T_P.
The advantage of analyzing $\mathrm{T_A}$ is that we need only to count the total number of proposals made – without regard to the name of the proposer at each stage.

New algorithm terminates with a stable marriage once each woman has received at least one proposal (for a "marriage").

The task to determine a good upper bound of T_A is a special case of the task to determine such a bound for so-called **Coupon Selection Problem** discussed next.

At the end we will get:

Theorem For any constant $c \in \Re$ and $m = n \ln n + cn$

$$\lim Pr[\,\mathrm{T_A} > m\,] = 1 - e^{-e^{-c}}$$

COUPON SELECTION PROBLEM

COUPON SELECTION PROBLEM

There are n types of coupons and at each time a coupon is chosen at random.

COUPON SELECTION PROBLEM

There are n types of coupons and at each time a coupon is chosen at random. The task is to determine for each $m \geq n$ the probability of having collected at least one of each of the n types of coupons in m trials.

COUPON SELECTION PROBLEM

There are n types of coupons and at each time a coupon is chosen at random. The task is to determine for each $m \geq n$ the probability of having collected at least one of each of the n types of coupons in m trials.

Elementary analysis Let X be a random variable the value of which is the number of trials required to collect at least one of each type of coupons.

COUPON SELECTION PROBLEM

There are n types of coupons and at each time a coupon is chosen at random. The task is to determine for each $m \geq n$ the probability of having collected at least one of each of the n types of coupons in m trials.

Elementary analysis Let X be a random variable the value of which is the number of trials required to collect at least one of each type of coupons.

Let $C_1, \ldots, C_X$ denote a sequence of trials.

COUPON SELECTION PROBLEM

There are n types of coupons and at each time a coupon is chosen at random. The task is to determine for each $m \geq n$ the probability of having collected at least one of each of the n types of coupons in m trials.

Elementary analysis Let X be a random variable the value of which is the number of trials required to collect at least one of each type of coupons.

Let $C_1, \ldots, C_X$ denote a sequence of trials.

The i-th trial C_i will be called success if the coupon selected in the trial C_i was not drawn in any of the first $i - 1$ selections.

COUPON SELECTION PROBLEM

There are n types of coupons and at each time a coupon is chosen at random. The task is to determine for each $m \geq n$ the probability of having collected at least one of each of the n types of coupons in m trials.

Elementary analysis Let X be a random variable the value of which is the number of trials required to collect at least one of each type of coupons.

Let $C_1, \ldots, C_X$ denote a sequence of trials.

The i-th trial C_i will be called success if the coupon selected in the trial C_i was not drawn in any of the first $i-1$ selections.

Sequence $C_1, \ldots, C_X$ will be divided into **epochs**. i-th epoch begins with the trial following the i-th success and ends with the trial which is $(i+1)$-th success.

COUPON SELECTION PROBLEM

There are n types of coupons and at each time a coupon is chosen at random. The task is to determine for each $m \geq n$ the probability of having collected at least one of each of the n types of coupons in m trials.

Elementary analysis Let X be a random variable the value of which is the number of trials required to collect at least one of each type of coupons.

Let $C_1, \ldots, C_X$ denote a sequence of trials.

The i-th trial C_i will be called success if the coupon selected in the trial C_i was not drawn in any of the first $i-1$ selections.

Sequence $C_1, \ldots, C_X$ will be divided into **epochs**. i-th epoch begins with the trial following the i-th success and ends with the trial which is $(i+1)$-th success.

Let $X_i, 0 \leq i < n$, be the number of trials in the i-th epoch. Then

$$X = \sum_{i=0}^{n-1} X_i$$

If p_i is the probability of the success on any trial of the i-th epoch then

$$p_i = \frac{n-i}{n}.$$

If p_i is the probability of the success on any trial of the i-th epoch then

$$p_i = \frac{n-i}{n}.$$

Random variable X_i is geometrically distributed, with the parameter p_i, and therefore its average value is $E[X_i] = \frac{1}{p_i} = \frac{n}{n-i}$ and its variance $V[X_i] = \sigma^2_{X_i} = \frac{1-p_i}{p_i^2} = \frac{ni}{(n-i)^2}$.

If p_i is the probability of the success on any trial of the i-th epoch then

$$p_i = \frac{n-i}{n}.$$

Random variable X_i is geometrically distributed, with the parameter p_i, and therefore its average value is $E[X_i] = \frac{1}{p_i} = \frac{n}{n-i}$ and its variance $V[X_i] = \sigma^2_{X_i} = \frac{1-p_i}{p_i^2} = \frac{ni}{(n-i)^2}$.

By the linearity of expectations we have:

$$\mathbf{E}[X] = \mathbf{E}\left[\sum_{i=0}^{n-1} X_i\right] = \sum_{i=0}^{n-1} \mathbf{E}[X_i] = \sum_{i=0}^{n-1} \frac{n}{n-i} = n\sum_{i=1}^{n} \frac{1}{i} = nH_n.$$

If p_i is the probability of the success on any trial of the i-th epoch then

$$p_i = \frac{n-i}{n}.$$

Random variable X_i is geometrically distributed, with the parameter p_i, and therefore its average value is $E[X_i] = \frac{1}{p_i} = \frac{n}{n-i}$ and its variance $V[X_i] = \sigma^2_{X_i} = \frac{1-p_i}{p_i^2} = \frac{ni}{(n-i)^2}$.

By the linearity of expectations we have:

$$\mathbf{E}[X] = \mathbf{E}\left[\sum_{i=0}^{n-1} X_i\right] = \sum_{i=0}^{n-1} \mathbf{E}[X_i] = \sum_{i=0}^{n-1} \frac{n}{n-i} = n\sum_{i=1}^{n}\frac{1}{i} = nH_n.$$

Since X_i are independent

$$\sigma^2_X = \sum_{i=0}^{n-1} \sigma^2_{X_i} = \sum_{i=0}^{n-1} \frac{ni}{(n-i)^2} = \sum_{i=0}^{n-1} \frac{n(n-i)}{i^2} = n^2\sum_{i=1}^{n}\frac{1}{i^2} - nH_n$$

If p_i is the probability of the success on any trial of the i-th epoch then

$$p_i = \frac{n-i}{n}.$$

Random variable X_i is geometrically distributed, with the parameter p_i, and therefore its average value is $E[X_i] = \frac{1}{p_i} = \frac{n}{n-i}$ and its variance $V[X_i] = \sigma^2_{X_i} = \frac{1-p_i}{p_i^2} = \frac{ni}{(n-i)^2}$.

By the linearity of expectations we have:

$$\mathbf{E}[X] = \mathbf{E}\left[\sum_{i=0}^{n-1} X_i\right] = \sum_{i=0}^{n-1} \mathbf{E}[X_i] = \sum_{i=0}^{n-1} \frac{n}{n-i} = n\sum_{i=1}^{n} \frac{1}{i} = nH_n.$$

Since X_i are independent

$$\sigma^2_X = \sum_{i=0}^{n-1} \sigma^2_{X_i} = \sum_{i=0}^{n-1} \frac{ni}{(n-i)^2} = \sum_{i=0}^{n-1} \frac{n(n-i)}{i^2} = n^2 \sum_{i=1}^{n} \frac{1}{i^2} - nH_n$$

Since $\lim_{n\to\infty} \sum_{i=1}^{n} \frac{1}{i^2} = \frac{\pi^2}{6}$ we have $\lim_{n\to\infty} \frac{\sigma^2_X}{n^2} = \frac{\pi^2}{6}$.

We show now that X unlikely deviates much from expectation

We show now that X unlikely deviates much from expectation

Let ε_i^r denote the event that a coupon of type i is not collected in the first r trials.

$$Pr[\varepsilon_i^r] = (1 - \frac{1}{n})^r \leq e^{-\frac{r}{n}} = n^{-\beta} \quad \textbf{for} \quad r = \beta n \ln n$$

We show now that X unlikely deviates much from expectation

Let ε_i^r denote the event that a coupon of type i is not collected in the first r trials.

$$Pr[\varepsilon_i^r] = (1 - \frac{1}{n})^r \leq e^{-\frac{r}{n}} = n^{-\beta} \quad \textbf{for} \quad r = \beta n \ln n$$

Therefore, for $r = \beta n \ln n$, we get

$$\mathbf{Pr}[X > r] = Pr\left[\cup_{i=1}^{n} \varepsilon_i^r\right] \leq \sum_{i=1}^{n} \mathbf{Pr}[\varepsilon_i^r] \leq \sum_{i=1}^{n} n^{-\beta} = n^{-(\beta-1)}$$

Next aim: To study the probability that X deviates from its expectation nH_n by the amount cn for any real c.

A TECHNICAL LEMMA and MAIN THEOREM

Lemma Let c be a real number and $m = n \ln n + cn$ for a positive integer n. Then, for any fixed k it holds

$$\lim_{n \to \infty} \binom{n}{k} (1 - \frac{k}{n})^m = \frac{e^{-ck}}{k!}.$$

MAIN THEOREM 1/4

Theorem Let the random variable X denote the number of trials for collecting each of the n types of coupons. Then for any $c \in \mathbf{R}$ and $m = n \ln n + cn$

$$\lim_{n\to\infty} Pr[X > m] = 1 - e^{-e^{-c}}$$

Proof Consider the event $\{X > m\} = \bigcup_{i=1}^{n} \varepsilon_i^m$. By the principle of the Inclusion-Exclusion

$$Pr\left[\bigcup_{i=1}^{n} \varepsilon_i^m\right] = \sum_{k=1}^{n} (-1)^{k+1} P_k^n \qquad (*)$$

where

$$P_k^n = \sum_{1\le i_1<\cdots<i_k\le n} Pr\left[\bigcap_{j=1}^{k} \varepsilon_{i_j}^m\right].$$

Let

$$S_k^n = P_1^n - P_2^n + P_3^n - \cdots + (-1)^{k+1} P_k^n$$

denote the partial sum formed by the first k terms in $(*)$.

MAIN THEOREM 1/4

Theorem Let the random variable X denote the number of trials for collecting each of the n types of coupons. Then for any $c \in \mathbf{R}$ and $m = n \ln n + cn$

$$\lim_{n\to\infty} Pr[X > m] = 1 - e^{-e^{-c}}$$

Proof Consider the event $\{X > m\} = \bigcup_{i=1}^{n} \varepsilon_i^m$. By the principle of the Inclusion-Exclusion

$$Pr\left[\bigcup_{i=1}^{n} \varepsilon_i^m\right] = \sum_{k=1}^{n} (-1)^{k+1} P_k^n \qquad (*)$$

where

$$P_k^n = \sum_{1 \le i_1 < \cdots < i_k \le n} Pr\left[\bigcap_{j=1}^{k} \varepsilon_{i_j}^m\right].$$

Let

$$S_k^n = P_1^n - P_2^n + P_3^n - \cdots + (-1)^{k+1} P_k^n$$

denote the partial sum formed by the first k terms in $(*)$. By Boole-Bonferroni inequalities

$$S_{2k}^n \le Pr\left[\bigcup_{i=1}^{n} \varepsilon_i^m\right] \le S_{2k+1}^n$$

REMAINDER

THE INCLUSION-EXCLUSION PRINCIPLE

Let $A_1, A_2, \ldots, A_n$ be events – not necessarily disjoint. The **Inclusion-Exclusion principle**, that has also a variety of applications, states that

$$
\begin{aligned}
Pr\left[\bigcup_{i=1}^{n} A_i\right] &= \sum_{i=1}^{n} Pr(A_i) - \sum_{i<j} Pr(A_i \cap A_j) + \sum_{i<j<k} Pr(A_i \cap A_j \cap A_k) - \\
&- \ \ldots \ + (-1)^{k+1} \sum_{i_1<i_2<\ldots<i_k} Pr\left[\bigcap_{j=1}^{k} A_{i_j}\right] \ \ldots \ + \\
&+ (-1)^{n+1} Pr\left[\bigcap_{i=1}^{n} A_i\right]
\end{aligned}
$$

SPECIAL CASES of THE INCLUSION-EXCLUSION PRINCIPLE

SPECIAL CASES of THE INCLUSION-EXCLUSION PRINCIPLE

"Markov"-type inequality - Boole's inequality or Union bound

$$Pr\left(\bigcup_i A_i\right) \leq \sum_i Pr(A_i)$$

SPECIAL CASES of THE INCLUSION-EXCLUSION PRINCIPLE

"Markov"-type inequality - Boole's inequality or Union bound

$$Pr\left(\bigcup_i A_i\right) \leq \sum_i Pr(A_i)$$

"Chebyshev"-type inequality

$$Pr\left(\bigcup_i A_i\right) \geq \sum_i Pr(A_i) - \sum_{i<j} Pr(A_i \cap A_j)$$

SPECIAL CASES of THE INCLUSION-EXCLUSION PRINCIPLE

"Markov"-type inequality - Boole's inequality or Union bound

$$Pr\left(\bigcup_i A_i\right) \leq \sum_i Pr(A_i)$$

"Chebyshev"-type inequality

$$Pr\left(\bigcup_i A_i\right) \geq \sum_i Pr(A_i) - \sum_{i<j} Pr(A_i \cap A_j)$$

Another proof of Boole's inequality:

Let us define $B_i = A_i - \bigcup_{j=1}^{i-1} A_j$. Then $\bigcup A_i = \bigcup B_i$. Since B_i are disjoint and for each i we have $B_i \subset A_i$ we get

$$Pr[\bigcup A_i] = Pr[\bigcup B_i] = \sum Pr[B_i] \leq \sum Pr[A_i]$$

MAIN THEOREM 2/4 - CONTINUATION

By symmetry, all the k-wise intersections of the events ε_i^m are equally likely, and therefore

$$P_k^n = \binom{n}{k} Pr\left[\bigcap_{i=1}^{k} \varepsilon_i^m\right].$$

Probability of the intersection of k events $\varepsilon_1^m, \ldots, \varepsilon_k^m$ is the probability of not collecting any of the first k coupons in m trials, namely $(1 - \frac{k}{n})^m$. Therefore $P_k^n = \binom{n}{k}\left(1 - \frac{k}{n}\right)^m$.

By the last Lemma, for $m = n \ln n + cn$

$$lim_{n\to\infty} P_k^n = \frac{e^{-ck}}{k\,!} = P_k \text{ — \{notation\}}.$$

Let us denote also:

$$S_k = \sum_{j=1}^{k}(-1)^{j+1}P_j = \sum_{j=1}^{k}(-1)^{j+1}\frac{e^{-cj}}{j\,!}. \quad (**)$$

The right hand side of $(**)$ consists precisely of k terms of the power series expansion of $f(c) = 1 - e^{-e^{-c}}$.

Hence

MAIN THEOREM 3/4

Therefore, for all $\varepsilon > 0$ there exists $k^* > 0$ such that for any $k > k^*$

$$|S_k - f(c)| < \varepsilon.$$

Since $\lim_{n\to\infty} P_k^n = P_k$, we have $\lim_{n\to\infty} S_k^n = S_k$. Equivalently, for all $\varepsilon > 0$ and all k, for all sufficiently large. n

$$|S_k^n - S_k| < \varepsilon$$

Thus, for all $\varepsilon > 0$ any fixed $k > k^*$, and n sufficiently large

$$|S_k^n - S_k| < \varepsilon, \quad |S_k - f(c)| < \varepsilon$$

$$\Longrightarrow |S_k^n - f(c)| = |S_k^n - S_k| + |S_k - f(c)| < 2\varepsilon$$

and

$$\left|S_{2k}^n - S_{2k+1}^n\right| < 4\varepsilon.$$

As a consequence

$$\left|Pr\left[\bigcup_{i=1}^{n} \varepsilon_i^m\right] - f(c)\right| < 4\varepsilon$$

and therefore

$$\lim_{n\to\infty} Pr\left[\bigcup_{i=1}^{n} \varepsilon_i^m\right] = f(c) = 1 - e^{-e^{-c}}$$

MAIN THEOREM 4/4

what implies

$$\lim_{n\to\infty} Pr[X > n(\ln n + c)] = 1 - e^{-e^{-c}}$$

Implications With extremely high probability, the number of trials, for collecting all n coupon types, lies in a small interval centered about its expected value.

A SUMMARY of the ANALYSIS of STABLE MARRIAGE PROBLEM

In case of the stable marriage problem of n men and women we have

- The worst case complexity (of the number of proposals) in n^2,
- The average case complexity is $\mathcal{O}(n \lg n)$.
- Deviation is small from the expected case.

APPENDIX

SIMILAR PROBLEMS

SIMILAR PROBLEMS

Generalised stable marriage problem A man (woman) may not be willing to marry some partners from the opposite sex and may prefer to stay single.

SIMILAR PROBLEMS

Generalised stable marriage problem A man (woman) may not be willing to marry some partners from the opposite sex and may prefer to stay single.

Stable roommate problem is similar to the stable marriage problem, but all participants belong to a single pool (Group).

SIMILAR PROBLEMS

Generalised stable marriage problem A man (woman) may not be willing to marry some partners from the opposite sex and may prefer to stay single.

Stable roommate problem is similar to the stable marriage problem, but all participants belong to a single pool (Group).

Hospitals-students(medical) problem This differs from the stable marriage problem that a women [hospital] can accept "proposals" from more than one man [student].

SIMILAR PROBLEMS

Generalised stable marriage problem A man (woman) may not be willing to marry some partners from the opposite sex and may prefer to stay single.

Stable roommate problem is similar to the stable marriage problem, but all participants belong to a single pool (Group).

Hospitals-students(medical) problem This differs from the stable marriage problem that a women [hospital] can accept "proposals" from more than one man [student].

Hospital-students problems with couples Similar problem as the above one, but among students can be couples that have to be assigned either to the same hospital or to a specific pair of hospitals chosen by couples.

1. Which of the numbers e^π and π^e, is larger, for the case that e is the basis of natural logarithms

EXERCISES

1. Which of the numbers e^{π} and π^{e}, is larger, for the case that e is the basis of natural logarithms
2. Hint 1: There exists one-line proof of the correct relation.