Chapter 5. 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 first show how so called *second moment method* can be used to determine thresholds of some events.
Threshold of an event $E_{p,n}$, that depends on a probability p and a size-parameter n,, is a number $t_{E,n}$ such that if $p < t_{E,n}$, then the probability of the event $E_{p,n}$ goes to 0 (if $n \to \infty$), and if $p > t_{E,n}$, then the probability of $E_{p,n}$ goes to 1.

We will then discuss, in various details, three important problems: Occupancy (Balls-into-Bins) problem, Stable marriage problem and Coupon selection problem.

VARIANCE of the SUM of RANDOM VARIABLES

If X_i, $i = 1, 2, \ldots, n$, are random variables, and

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

then

$$\begin{aligned}
\mathrm{VAR}[X] &= \mathbf{E}[X^2] - (\mathbf{E}[X])^2 && (1)\\
&= \sum_{i=1}^{n} \mathbf{E}[X_i^2] + \sum_{i \neq j} \mathbf{E}[X_i X_j] - \sum_{i=1}^{n} (\mathbf{E}[X_i])^2 - \sum_{i \neq j} \mathbf{E}[X_i]\mathbf{E}[X_j] && (2)\\
&= \sum_{i=1}^{n} (\mathbf{E}[X_i^2] - (\mathbf{E}[X_i])^2) + \sum_{i \neq j} (\mathbf{E}[X_i X_j] - \mathbf{E}[X_i]\mathbf{E}[X_j]) && (3)\\
&= \sum_{i=1}^{n} \mathrm{VAR}[X_i] + \sum_{i \neq j} (\mathbf{E}[X_i X_j] - \mathbf{E}[X_i]\mathbf{E}[X_j]) && (4)
\end{aligned}$$

$$\mathbf{E}[X_i X_j] - \mathbf{E}[X_i]\mathbf{E}[X_j] = \mathbf{E}[(X_i - \mathbf{E}[X_i])(X_j - \mathbf{E}[X_j])]$$

is called **Covariance** of X_i and X_j and is denoted by $\mathbf{Cov}(X_i, X_j)$.

THRESHOLDS – BASICS

Many interesting/important graph properties have **threshold functions in the family of random graphs $G_{n,p}$**.

Definition The family of random graphs $G_{n,p}$ is defined to be the family of all graphs on n nodes where each edge is chosen with probability p.

Threshold for a property P of graphs from $G_{n,p}$ is such a value p^* that if $p < p^*$, then the probability that a graph sampled from $G_{n,p}$ has the property P goes to 0, and if $p > p^*$, then such probability goes to 1 (for $n \to \infty$).

Useful fact Let X be a non-negative random variable with $\mathbf{E}[X] = \mu$ and $\mathrm{VAR}[X] = \sigma^2$. By Chebyshev's inequality

$$Pr[|X - \mu| \geq \lambda\sigma] \leq \frac{1}{\lambda^2}.$$

By choosing $\lambda = \frac{\mu}{\sigma}$, we get

$$Pr[X = 0] \leq Pr[(X = 0) \cup (X \geq 2\mu)] = Pr[|X - \mu| \geq \mu] \leq \frac{\sigma^2}{\mu^2} = \frac{\mathrm{VAR}[X]}{E[X]^2}. \tag{5}$$

Therefore, if $\mathrm{VAR}[X] = o(\mathbf{E}[X]^2)$, we get $Pr[X = 0] \to 0$.

CONDITIONS WHEN PROBABILITY GOES TO 0

Let $\mathcal{E}_1, \ldots, \mathcal{E}_n$ be events having the same probability p and let X_i be the indicator variable for $\mathcal{E}_i$. Finally, let $X = \sum_{i=1}^{n} X_i$.

Denote by $i \backsim j$ the case that events $\mathcal{E}_i$ and $\mathcal{E}_j$ are dependent and $i \neq j$. By definition

$$\mathrm{VAR}[X] = \sum_{i=1}^{n} \mathrm{VAR}[X_i] + \sum_{i \backsim j} \mathrm{Cov}(X_i, X_j),$$

where

$$\mathrm{Cov}(X_i, X_j) = \mathbf{E}[X_i X_j] - \mathbf{E}[X_i]\mathbf{E}[X_j] \le \mathbf{E}[X_i X_j] = Pr[X_i \cap X_j].$$

Since X_i are Bernoulli random variables, it holds

$$\mathrm{VAR}[X_i] = p(1-p) \le p = \mathbf{E}[X_i] \Longrightarrow \mathrm{VAR}[X] \le \sum_{i=1}^{n} \mathbf{E}[X_i] + \sum_{i \backsim j} Pr[X_i \cap X_j]$$

By (5), and using the notation $\Delta^* = \sum_{i \backsim j} Pr[X_i \cap X_j]$, we get

$$Pr[X = 0] \le \frac{\mathrm{VAR}[X]}{\mathbf{E}[X]^2} = \frac{\mathbf{E}[X] + \Delta^*}{\mathbf{E}[X]^2} \le \frac{1}{\mathbf{E}[X]} + \frac{\Delta^*}{\mathbf{E}[X]^2},$$

Therefore, if $\mathbf{E}[X] \to \infty$ and $\Delta^* = o(\mathbf{E}[X]^2)$, then $Pr[X = 0] \to 0$.

THRESHOLD FUNCTION for CLIQUE

For a graph G let $SLC(G)$ be the size of the largest clique of G.
Theorem Event $SLC(G_{n,p}) \geq 4$ has a threshold (function) $p^* = n^{-2/3}$.

Proof Let us consider a random $G_{n,p}$ graph.
For any set S of 4 vertices of $G_{n,p}$ let A_S be the event that S forms a clique in G and let X_S be the indicator variable of A_S.

Clearly, $\boldsymbol{Pr}[A_S] = p^6$ for any set S of 4 vertices.

If $X = \sum_{S,|S|=4} X_S$, then $\mathbf{E}[X] = \binom{n}{4} \cdot p^6 \approx \frac{n^4 p^6}{24}$. By Markov's inequality

$$\boldsymbol{Pr}[X \geq 1] \leq \frac{\mathbf{E}[X]}{1}$$

and therefore if $p \ll n^{-2/3}$, then $\mathbf{E}[X] = o(1)$ and therefore

$$\boldsymbol{Pr}[SLC(G_{n,p}) \geq 4] = \boldsymbol{Pr}[X \geq 1] \rightarrow 0.$$

Let now $p \gg n^{-2/3}$.

Observe first that then $\mathbf{E}[X] \to \infty$. Secondly, observe that two cliques are dependent iff they intersect in at least two vertices.

Let now S and T be any two sets of four vertices:

1. **If $|S \cap T| = 2$, then $\boldsymbol{Pr}[A_T|A_S] = p^5$. (Observe that for any fixed set S there are $\mathcal{O}(n^2)$ such sets T.)**
2. **If $|S \cap T| = 3$, then $\boldsymbol{Pr}[A_T|A_S] = p^3$. (Observe that for any fixed set S there are $\mathcal{O}(n)$ such sets T.)**

Therefore, for Δ^* from previous slide:

$$\Delta^* = \sum_{T \backsim S} \boldsymbol{Pr}[A_T|A_S] \approx np^3 + n^2p^5.$$

Hence, if $p \gg n^{-2/3}$, then $\mathbf{E}[X] \to \infty$, $\Delta^* = o(\mathbf{E}[X])$, and therefore

$$\boldsymbol{Pr}[SLC(G_{n,p}) < 4] = \boldsymbol{Pr}[X = 0] \to 0.$$

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 bins look like?

Subproblem 1: How many of the bins will be empty? For a given k what is the probability that k boxes will be empty?

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

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

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**.

An important inequality

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

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

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).

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

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

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

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

BASIC COROLLARIES

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

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 numbers in the expected time $\mathcal{O}(n)$.**

BUCKET SORT ALGORITHM - STAGE 1

Stage 1. We place numbers into n buckets in such a way that all numbers whose first m bits represent number j 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, this 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

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 $B(n, \frac{1}{n})$, by using the result from 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

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

If there are k such people in a room than probability that each of them has birthday in a different day is

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

what equals to

$$\frac{\binom{365}{k}}{365^k \cdot k!}$$

using the inequality $1-\frac{j}{n} \approx e^{-j/n}$ for j small - comparing to n - we have

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

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

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

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

VARIATIONS on BIRTHDAY PARADOX

It is well known 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 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 (so that 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% $(1 - e^{\lambda})\%$**.**

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

MARKOV and CHEBYSHEV INEQUALITIES AGAIN

There are general inequalities on tail probabilities that are very often much useful.
If X is a discrete random variable and $f(x)$ a real function, then

$$\mathbf{E_X}[f(x)] = \sum_x f(x) Pr[X = x].$$

Theorem (Markov Inequality) Let Y be a random variable with only non-negative values. Then for all $t \in \mathbf{R}^+$

$$Pr[Y \geq t] \leq \frac{\mathbf{E}[Y]}{t},$$

or, equivalently,

$$Pr[Y \geq k\mathbf{E}[Y]] \leq \frac{1}{k}.$$

Proof For a real t let us define a function $f_t(y) = 1$ if $y \geq t$ and 0 otherwise. Then

$$Pr[Y \geq t] = \mathbf{E}_Y[f_t(y)].$$

Since $f_t(y) \leq \frac{y}{t}$ for all y,

$$\mathbf{E}[f_t(y)] \leq \mathbf{E}\left[\frac{y}{t}\right] = \frac{\mathbf{E}[y]}{t}.$$

A better bound on the tail probability can be obtained using more information about the distribution of the random variable.

Theorem (Chebyshev's Inequality) Let X be a random variable with expectation μ_X and standard deviation σ_X. Then for all $t \in \mathbf{R}^+$

$$Pr[|X - \mu_X| \geq t\sigma_X] \leq \frac{1}{t^2}.$$

Proof Note that $Pr[|X - \mu_X| \geq t\sigma_X] = Pr[(X - \mu_X)^2 \geq t^2\sigma_X^2]$

The random variable $Y = (X - \mu_X)^2$ has expectation σ_X^2. Markov inequality applied to Y bounds this probability from above by $\frac{1}{t^2}$.

Another version of Chebyshev's Inequality. For any $a > 0$

$$Pr(|X - \mu_X| \geq a) \leq \frac{\sigma_X^2}{a^2}$$

EXAMPLE

Let $X = \sum_{i=1}^{n} X_i$ denote the number of heads in the n coin tossing - with $X_i = 1$ (0) if the ith coin flip is heads (tails).

Let us try to bound probability of obtaining more than $3n/4$ heads in a sequence of n fair coin flips.

It holds $\mathbf{E}[X_i] = \frac{1}{2}$, $\mathbf{E}[X] = \frac{n}{2}$ and $\mathbf{VAR}[X_i] = \frac{1}{4}$, $\mathbf{VAR}[X] = \frac{n}{4}$,

Markov's inequality gives

$$Pr(X > 3n/4) \leq \frac{\mathbf{E}[X]}{3n/4} = \frac{n/2}{3n/4} = \frac{2}{3}$$

Chebyshev's inequality gives

$$Pr(X \geq 3n/4) \leq Pr(|X - \mathbf{E}[X]| \geq n/4) \leq \frac{\mathbf{VAR}[X]}{(n/4)^2} = \frac{n/4}{(n/4)^2} = \frac{4}{n}$$

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

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

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

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

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 of his/her preferences.

Task: Marry men and women together such 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, . . .
and n women a, b, c, . . .

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: $A - a$, $B - b$, $C - c$, $D - d$.

COMMENT

- **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 anlysis of algorithms, CRM Proceedings and Lecture Notes,**
- **Algorithms to deal with this type of problems are used**
  - **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 exist.

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:

$$M_1 : W_2W_1W_3, \quad M_2 : W_3W_2W_1, \quad M_3 : W_1W_3W_2$$
$$W_1 : M_2M_1M_3, \quad W_2 : M_3M_2M_1, \quad W_3 : M_1M_3M_2$$

There are three stable solutions:

- Men get their first choice and women the third one:
$$M_1W_2, M_2W_3, M_3W_1$$
- All get their second choice
$$M_1W_1, M_2W_2, M_3W_3$$
- Women get their first choice and men the third one:
$$M_1W_3, M_2W_1, M_3W_2$$

A naive, but bad, 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 bad because a loop can occur.

EXAMPLE 1

For choices:

$$M_1 : W_3W_2W_4W_1 \quad M_2 : W_2W_1W_3W_4$$
$$M_3 : W_2W_4W_1W_3 \quad M_4 : W_3W_1W_4W_2$$

and

$$W_1 : M_1M_2M_4M_3 \quad W_2 : M_3M_1M_4M_2$$
$$W_3 : M_3M_2M_4M_1 \quad W_4 : M_2M_1M_3M_4$$

Successful development of marriages;

$$\mathbf{M_1}W_1 \quad M_2\mathbf{W_2} \quad M_3W_3 \quad M_4W_4 - -\text{unstable}$$
$$M_1\mathbf{W_2} \quad M_2W_1 \quad \mathbf{M_3}W_3 \quad M_4W_4 - -\text{unstable}$$
$$M_1\mathbf{W_3} \quad M_2W_1 \quad M_3W_2 \quad \mathbf{M_4}W_4 - -\text{unstable}$$
$$M_1W_4 \quad M_2W_1 \quad M_3W_2 \quad M_4W_3 - -\text{!stable!}$$

EXAMPLE 2

For choices:

$$M_1 : W_2W_1W_3 \quad M_2 : \text{arbitrary} \quad M_3 : W_1W_2W_3$$

and

$$W_1 : M_1M_3M_2 \quad W_2 : M_3M_1M_2 \quad W_3 : \text{arbitrary}$$

we have the following cyclic development of marriages

$$\begin{array}{c}
\mathbf{M_1}W_1 \quad M_2\mathbf{W_2} \quad M_3W_3 \\
M_1\mathbf{W_2} \quad M_2W_1 \quad \mathbf{M_3}W_3 \\
M_1W_3 \quad M_2\mathbf{W_1} \quad \mathbf{M_3}W_2 \\
\mathbf{M_1}W_3 \quad M_2W_2 \quad M_3\mathbf{W_1} \\
M_1W_1 \quad M_2W_2 \quad M_3W_3
\end{array}$$

EXAMPLE 3

For choices

$$M_1 : W_1W_2W_3W_4W_5 \quad M_2 : W_2W_3W_4W_5W_1 \quad M_3 : W_3W_4W_5W_1W_2$$
$$M_4 : W_4W_5W_1W_2W_3 \quad M_5 : W_5W_1W_2W_3W_4W_5$$

and

$$W_1 : M_2M_3M_4M_5M_1 \quad W_2 : M_3M_4M_5M_1M_2 \quad W_3 : M_4M_5M_1M_2M_3$$
$$W_4 : M_5M_1M_2M_3M_4 \quad W_5 : M_1M_2M_3M_4M_5$$

we have exactly 5 stable marriages

$$M_1W_1 \quad M_2W_2 \quad M_3W_3 \quad M_4W_4 \quad M_5W_5$$
$$M_1W_2 \quad M_2W_3 \quad M_3W_4 \quad M_4W_5 \quad M_5W_1$$
$$M_1W_3 \quad M_2W_4 \quad M_3W_5 \quad M_4W_1 \quad M_5W_2$$
$$M_1W_4 \quad M_2W_5 \quad M_4W_1 \quad M_4W_2 \quad M_5W_3$$
$$M_1W_5 \quad M_2W_1 \quad M_4W_2 \quad M_5W_3 \quad M_5W_4$$

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

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 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 reject him yet. The woman W then decides whether to accept his proposal or to reject it.

W accepts the proposal if

- she is not yet married or
- she likes M more than the 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

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

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-proposl algorithm does not depend on the order men are chosen to make their proposals.

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

- National residency matching program
- Dental resiencies and medical specialtes in the USA, Canada and parts of UK
- National university entrance exam in Iran
- Placement of Canadian lawyers in Ontario and Alberta
- Sorority rush
- Matching of new reform rabbis to their first congregation
- Assignment of students to high-schools in NYC

STABLE HUSBANDS

The stable husband of a woman, with respect to a given rankings, is a man she can be maried 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 matchins defined by these rankings, with probability approaching 1 as $n \longrightarrow \infty$, if ϵ is any positive constant.

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

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 present at first a very simple technique that allows to analyse randomized algorithms with seemingly very complex behaviour. The technique will be first illustrated on a simple card game.

1. Game ”Clock Solitaire”

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, . . . , 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

On the first move a card is drawn from the pile labeled 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.

We win the game if, on termination, all 52 cards have been drawn. In all other cases we lose 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	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	2
4	5	6	7	8	9	10	J	Q	K	A	2	3

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, $\frac{1}{13}$.

> The idea of the *Principle of Deferred Decisions* is not to assume that the entire set of random choices is made in advance, rather, that **at each step of the algorithm we fix only that random choice that needs to be revealed at that step**.

ANALYSIS of a RANDOMIZED VERSION of PROPOSAL ALGORITHM

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.

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

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

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").

To 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 $m = n \ln n + cn$

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

COUPON SELECTION PROBLEM

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

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

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 is unlikely to deviate much from its 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

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

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

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

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

Let

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

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

By Boole-Bonferroni inequalities

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

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

$P_k^n = \binom{n}{k} (1 - \frac{k}{n})^m.$

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

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

Let us denote also:

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

The right hand side of $(**)$ consists precisely of k terms of the power series expansion of $f(c) = 1 - e^{-e^{-c}}$.
Hence
$$\lim_{k\to\infty} S_k = f(c).$$
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
$$|S_k^n - S_k| < \varepsilon, \quad |S_k - f(c)| < \varepsilon$$
$$\implies |S_k^n - f(c)| = |S_k^n - S_k| + |S_k - f(c)| < 2\varepsilon$$
and
$$|S_{2k}^n - S_{2k+1}^n| < 4\varepsilon.$$
As a consequence
$$\left| Pr\left[\bigcup_{i=1}^{n} \varepsilon_i^m\right] - f(c)\right| < 4\varepsilon$$

and therefore

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

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

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.

A COUPON SELECTION PROBLEM APPLICATION

Let packets are sent in a stream from a source node to a destination node along a fixed path of routers.

Let us assume that the destination node would like/need to know which routers the stream of packets has passed through.

Let us assume that each packet can store, randomly chosen, the name of one of the routers it goes through.

Determining all the routers on the path is like a coupon collector's problem.

This means that if there are n routters on the path then the expected number of packets that need to be sent so the destination node knows all routers is $nH(n)$.

EXERCISES

rblack

1. What is larger, e^{π} or π^{e}, for the basis e of natural logarithms