

**Zadání a řešení testu z informatiky a zpráva  
o výsledcích přijímacího řízení do magisterského  
navazujícího studia od podzimu 2016**

**Zpráva o výsledcích přijímacího řízení  
do magisterského navazujícího studia od podzimu 2016**

|  |       |
|--|-------|
| Počet podaných přihlášek   | 398   |
| Počet přihlášených uchazečů  | 357   |
| Počet uchazečů, kteří splnili podmínky přijetí   | 256   |
| Počet uchazečů, kteří nesplnili podmínky přijetí   | 101   |
| Počet uchazečů přijatých ke studiu, bez uvedení počtu uchazečů přijatých ke studiu až na základě výsledku přezkoumání původního rozhodnutí | 256   |
| Počet uchazečů přijatých celkem  | 256   |
| Percentil pro přijetí  | 10,00 |

**Základní statistické charakteristiky**

|   | Matematika | Informatika | Celkem |       |
|---|------------|-------------|--------|-------|
| Počet otázek  | 25         | 30          | 55     |       |
| Počet uchazečů, kteří se zúčastnili přijímací zkoušky | 192        | 193         | 193    |       |
| Nejlepší možný výsledek                               | 25.00      | 30.00       | 55.00  |       |
| Nejlepší skutečně dosažený výsledek                   | 25.00      | 25.75       | 47.75  |       |
| Průměrný výsledek                                     | 14.95      | 12.61       | 27.49  |       |
| Medián  | 16.25      | 13.00       | 28.75  |       |
| Směrodatná odchylka                                   | 5.52       | 5.06        | 9.69   |       |
|   | Percentil  |             |        |       |
| Decilové hranice výsledku *                           | 10         | 7.50        | 5.75   | 14.50 |
|   | 20         | 10.25       | 8.00   | 18.75 |
|   | 30         | 12.25       | 10.00  | 23.00 |
|   | 40         | 14.25       | 12.00  | 26.00 |
|   | 50         | 16.25       | 13.00  | 28.75 |
|   | 60         | 17.00       | 14.00  | 31.25 |
|   | 70         | 18.25       | 15.50  | 33.50 |
|   | 80         | 19.75       | 16.50  | 35.75 |
|   | 90         | 21.25       | 19.00  | 39.50 |

\* Decilové hranice výsledku zkoušky vyjádřené d1, d2, d3, d4, d5, d6, d7, d8, d9 jsou hranice stanovené tak, že rozdělují uchazeče seřazené podle výsledku zkoušky do stejně velkých skupin, přičemž d5 je medián.

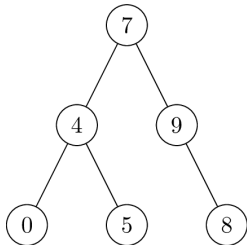
# Přijímací zkouška - Informatika

| Jméno a příjmení - pište do okénka | Číslo přihlášky | Číslo zadání |
|------------------------------------|-----------------|--------------|
|                                    |                 | 31           |

## Algoritmizace a datové struktury

- 1** Uvažujme orientovaný graf a dva jeho vrcholy  $u$  a  $v$  takové, že z  $u$  vede cesta do  $v$ . Které z následujících tvrzení je pravdivé?
- A Ani jedno ze zbylých tvrzení není pravdivé.
  - B Při prohledávání grafu do hloubky bude vždy vrchol  $u$  poprvé navštíven dříve než vrchol  $v$ .
  - C Při prohledávání grafu do šířky bude vždy vrchol  $u$  poprvé navštíven dříve než vrchol  $v$ .
  - \*D Pro každý vrchol  $s$ , ze kterého vede cesta do  $u$ , platí, že z něj vede cesta i do  $v$ .
  - E Pro každý vrchol  $s$ , ze kterého vede cesta do  $v$ , platí, že z něj vede cesta i do  $u$ .
- 
- 2** Mějme minimovou binární haldy (binary min heap), která obsahuje  $n$  prvků. Které z uvedených tvrzení **neplatí**?
- A Nalezení zadaného prvku v haldě má časovou složitost  $O(n)$ .
  - B Nalezení minima z prvků v haldě má časovou složitost  $O(\log n)$ .
  - C Odebrání minima z prvků v haldě má časovou složitost  $O(\log n)$ .
  - \*D Odebrání minima z prvků v haldě má časovou složitost  $O(1)$ .
  - E Nalezení minima z prvků v haldě má časovou složitost  $O(1)$ .
- 
- 3** Které z následujících tvrzení je pravdivé?
- A Vkládání prvků do binárního vyhledávacího stromu má časovou složitost vždy  $O(\log n)$ .
  - B Datová struktura hašovací tabulka je vhodná pro řazení posloupností čísel.
  - C Mezi binární vyhledávací stromy patří AVL stromy a binární haldy.
  - D Vstupní podmínka algoritmu popisuje vstupy, pro které má algoritmus optimální časovou složitost.
  - \*E Funkce  $n * \log(n)$  roste asymptoticky stejně rychle jako funkce  $\log(n!)$ .

**4**



Uvažujme strom uvedený výše. Kterou z následujících operací bychom s tímto stromem museli provést, aby byl výsledkem binární vyhledávací strom?

- A Žádnou, uvedený strom je binární vyhledávací strom.
- B Přesunout hodnotu 9 do kořene, hodnotu 7 do pravého listu a hodnotu 8 přesunout do pravého vrcholu v prostředním patře.
- C Přehodit hodnoty ve vrcholech 7 a 9.
- \*D Odstranit vrchol 8.
- E Přehodit hodnoty ve vrcholech 7 a 8.

- 5** Předpokládejme hašovací tabulku (hashtable) s řešením kolizí pomocí zřetěžených seznamů (separate chaining with linked lists). Velikost hašovací tabulky je 7, hašovací funkce je  $h(x) = (3 \cdot x + 5) \bmod 7$ . Na začátku výpočtu je tabulka prázdná, následně do ní postupně vkládáme hodnoty 12, 2, 5, 7, 9, 1. Které z uvedených tvrzení platí pro hašovací tabulku po tomto vkládání?
- A Jedna z položek tabulky obsahuje zřetěžený seznam s prvky 7 a 12.
  - B Nejdelší zřetěžený seznam v tabulce má délku 6 prvků.
  - \*C Ani jedno ze zbylých tvrzení není pravdivé.
  - D Jedna z položek tabulky obsahuje zřetěžený seznam s prvky 2 a 5.
  - E Nejdelší zřetěžený seznam v tabulce má délku 3 prvky.

## Počítačové systémy

- 6** Nechtě běh procesů na procesoru plánujeme algoritmem SJF (Shortest Job First) v preemptivní variantě (s předběháním). Požadavek na proces P1 vzniká v čase 0 a proces potřebuje 6 jednotek procesoru, požadavek na proces P2 vzniká v čase 2 a proces potřebuje 3 jednotky procesoru, požadavek na proces P3 vzniká v čase 2 a proces potřebuje 2 jednotky procesoru, požadavek na proces P4 vzniká v čase 3 a proces potřebuje 2 jednotky procesoru. Plánování procesů začíná v čase 0. Jaký proces bude běžet na procesoru v čase 7?
- A žádný z těchto procesů
  - \*B P2
  - C P1
  - D P4
  - E P3
- 7** Nutnou podmínkou uváznutí (deadlock) je:
- A možnost souběžného využití jedné instance zdroje alespoň dvěma vlákny.
  - \*B ponechání si již alokovaného zdroje a čekání na další zdroj (tzv. hold and wait).
  - C existence alespoň dvou instancí každého zdroje.
  - D nutnost využití alespoň 3 různých zdrojů.
  - E využití semaforu.
- 8** Každé vlákno v procesu má:
- A své zámky souborů.
  - B svou vlastní tabulku otevřených souborů.
  - C své obslužné rutiny pro ovladače signálů.
  - \*D svůj vlastní zásobník.
  - E svůj vlastní adresový prostor, který může pomocí IPC sdílet s jinými vlákny.
- 9** Dlouhodobý (strategický) plánovač v operačním systému:
- A musí být rychlý, neboť jej typicky spouštíme velice často.
  - B vybírá z připravených procesů jeden pro běh na procesoru.
  - C mění stav procesu/vlákná z čekající/blokovaný na připravený.
  - D rozhoduje, kdy který spuštěný proces odložit na disk.
  - \*E rozhoduje, kdy který program (úlohu) spustit.
- 10** Které číslo ve dvojkové soustavě je ekvivalentem čísla vyjádřeného v desítkové soustavě jako 23456?
- A 0000 1010 1011 0101
  - \*B 0101 1011 1010 0000
  - C 0111 1010 1010 1000
  - D 55640
  - E 1010 0100 0101 1111

## Programování

- 11** Rozhodněte, které z uvedených tvrzení je v běžných OOP jazycích (C++, Java, C#) obecně platné:
- A** Rozdíl mezi třídou a objektem je v tom, že objekty mohou dědit od tříd, ale naopak nikoliv.
  - B** Při časné vazbě (early binding; volání nevirtuální metody) se o tom, která metoda se přesně zavolá, rozhoduje až za běhu.
  - C** Pokud je metoda třídy statická, znamená to, že má na rozdíl od ostatních metod přístup i k soukromým atributům třídy.
  - \*D** Pokud třída B dědí od třídy A, pak má přístup ke všem atributům třídy A, které byly deklarovány jako public nebo protected.
  - E** Žádné z uvedených tvrzení není platné.
- 

**12** `read(n)`  
`sum = 0`  
`k = n`  
`XXX {`  
    `sum = sum + (n mod 10)`  
    `n = n div 10`  
`}`  
`print sum`

Nechť mod představuje zbytek po dělení a div je celočíselné dělení. Předpokládejme, že na vstupu výše uvedeného pseudokódu je kladné celé číslo  $n$ . Co musíme doplnit na místo označené "XXX", aby výše uvedený pseudokód dal na výstup ciferný součet čísla  $n$ ?

- A** `for i = 1 to 10`
  - B** `for i = 1 to k`
  - C** Žádná z uvedených možností není správná.
  - D** `while n > sum`
  - \*E** `while n > 0`
- 

- 13** Rozhodněte, která z uvedených tvrzení I, II a III jsou pravdivá (pro běžné jazyky typu C++, Java, C#). Vyberte takovou možnost, která obsahuje právě všechna pravdivá tvrzení (a neobsahuje žádné nepravdivé tvrzení).

I. Lokální proměnné funkcí jsou alokovány na zásobníku. Po opuštění funkce jsou automaticky dealokovány.

II. Paměť alokovaná pomocí operátoru new je alokována na haldě.

III. Není-li výjimka zachycena ve funkci, ve které je vyhozena, způsobí vždy pád programu.

- A** I
  - \*B** I, II
  - C** I, II, III
  - D** II, III
  - E** I, III
-

**14** function foo(integer n)

```
begin
    print n
    if n > 0 then
        print "*"
        foo(n-1)
    end if
    print n
end
```

```
program main()
begin
    foo(3)
end
```

Předpokládejte, že funkce print neukončuje výstup koncem řádku. Jaký bude výstup programu?

- A 3\*2\*1\*00
- \*B 3\*2\*1\*00123
- C 3\*2\*1\*123
- D Uvedený kód bude cyklit a nezastaví.
- E 3\*2\*1

**15** Které z následujících tvrzení **neplatí**?

- A Rekurzivní volání funkcí je typicky implementováno pomocí zásobníku.
- B Při volání referencí se změna parametru uvnitř funkce projeví i navenek.
- \*C Při volání hodnotou se změna parametru uvnitř funkce projeví i navenek.
- D Příkladem jazyka logického programování je Prolog.
- E Líná vyhodnocovací strategie ve funkcionálním programování umožňuje pracovat s nekonečnými seznamy.

## Počítačové sítě

**16** TCP protokol poskytuje tyto služby:

- A směrování dat se zajištěním kvality služby.
- B řízení toku a řízení zahlcení, kontrola chyb v aktivních prvcích.
- C podpora datových toků pro aplikace pracující v reálném čase.
- \*D tvorba bloků dat opatřených hlavičkou, jejich číslování, správa spojení, řízení chyb, řízení toku a řízení zahlcení.
- E směrování dat se zajištěním kvality služby, řízení toku a řízení zahlcení.

**17** Nejvyšší dosažitelná rychlost přenosu dat (bity/sekundu) v komunikačním kanálu je určena:

- A způsobem kódování dat do signálu v komunikačním kanálu.
- \*B šířkou intervalu frekvencí složkových signálů reprezentujících signál přenášející data a poměrem energie vysílání a škodlivé, nežádoucí energie přítomné v přenosovém médiu.
- C šířkou intervalu frekvencí složkových signálů reprezentujících signál přenášející data.
- D šířkou intervalu frekvencí složkových signálů reprezentujících signál přenášející data a způsobem kódování dat do tohoto signálu.
- E způsobem kódování dat do signálu v komunikačním kanálu a objemy dat, které jsou schopné vysílače a přijímače zpracovat za jednotku času.

- 18** Směrovací protokoly podle vektoru cesty (Path vector) pro směrování mezi autonomními systémy:
- A vyžadují, aby všechny autonomní systémy používaly stejnou metriku.
  - B vycházejí ze směrovacích protokolů podle stavu linky (Link state) a umožňují definici pravidel (politik).
  - C vycházejí ze směrovacích protokolů podle vektoru vzdálenosti (Distance vektor) a preferují vždy nejkratší cesty.
  - \*D vycházejí ze směrovacích protokolů podle vektoru vzdálenosti (Distance vektor), umožňují definici pravidel (politik) a preferují nejkratší cesty, pokud to pravidla dovolí.
  - E vyměňují si informaci o celých cestách, ale neumožňují detekci cyklů.
- 

- 19** Zajištění síťových služeb přepínáním paketů:
- \*A je realizováno zasíláním nezávislých datových jednotek (paketů) a je základem nespojovaných sítí.
  - B zajišťuje spojovanou službu bez potřeby paketizace.
  - C je základem spojovaných (connection oriented) sítí.
  - D ustavuje přímé fyzické spojení mezi odesílajícím a příjemcem.
  - E je realizováno zasíláním nezávislých datových jednotek (paketů) po ustavených fyzických spojích.
- 

- 20** Při všesměrovém bezdrátovém vysílání se vysílaná energie šíří od vysílače všemi směry. Přijímaná energie příjemcem:
- \*A se zvyšováním vzdálenosti příjemce od vysílače se snižuje kvadraticky.
  - B se zvyšováním vzdálenosti příjemce od vysílače se snižuje exponenciálně.
  - C se zvyšováním vzdálenosti příjemce od vysílače se snižuje lineárně.
  - D je shodná s vysílanou energií bez ohledu na vzdálenost příjemce od vysílače.
  - E se zvyšováním vzdálenosti příjemce od vysílače se snižuje kubicky.
- 

## Databázové systémy

---

- 21** Uvažujte následující relace z databáze banky: *zákazník*(*rč*, *jméno*, *adresa*), *účet*(*č*, *úctu*, *rč*, *zůstatek*). Předpokládejte, že obě relace jsou neprázdné, indexy existují pouze pro primární klíče. Atribut *účet.rč* je cizí klíč do relace *zákazník*. Z příkazů uvedených níže vyberte ten, který **nezrychlí** zpracování SQL dotazu
- ```
SELECT * FROM zákazník, účet WHERE zákazník.rč = účet.rč AND zůstatek = 0;
```
- A DELETE FROM účet WHERE zůstatek <> 0;
  - B CREATE INDEX zrychli ON účet(zůstatek);
  - C DELETE FROM účet WHERE zůstatek = 0;
  - D ALTER TABLE zákazník DROP adresa;
  - \*E CREATE INDEX zrychli ON zákazník(jméno);
- 

- 22** Druhá normální forma (2NF) vyžaduje:
- A aby všechny atributy, které jsou součástí libovolného cizího klíče, byly atomické.
  - B aby všechny závislosti byly atomické a aby zároveň platila i první normální forma (1NF).
  - C aby pro všechny atributy platilo pravidlo ACID.
  - D aby neexistoval atribut, který je tranzitivně závislý na některém kandidátním klíči.
  - \*E aby neexistoval atribut, který je závislý na části některého kandidátního klíče.
-

**23** Máme relaci  $r(A, B, C, D, E, F, G, H, I)$  a následující množinu funkčních závislostí:

$A, B, C, D \rightarrow E, F$

$A, C, D, E \rightarrow B$

$E \rightarrow G$

$A, E \rightarrow H$

$A, B \rightarrow I$

Co můžeme tvrdit o množině atributů  $\{A, B, C, G, H, I\}$ ?

- A Je to superklíč, ale není to kandidátní klíč.
- B Je to jediný primární klíč.
- C Je to superklíč i kandidátní klíč.
- \*D Není to ani kandidátní klíč, ani superklíč.
- E Je to kandidátní klíč, ale není to superklíč.

**24** Předpokládejme, že existuje relace *zamestnanec* mající relační schéma *zamestnanec*(*ID*, *jmeno*, *plat*). Doménou atributů *ID* a *plat* jsou přirozená čísla, doménou atributu *jmeno* jsou řetězce délky maximálně 10 znaků. Který z následujících výrazů je správným příkladem takové relace v relační algebře?

- A  $zamestnanec = \{1, 'Jan', 1000, 2, 'Jana', 2000, 3, 'Jitka', 3000\}$
- \*B  $zamestnanec = \{(1, 'Jan', 1000), (2, 'Jana', 2000), (3, 'Jitka', 3000)\}$
- C  $zamestnanec = (1, 'Jan', 1000, 2, 'Jana', 2000, 3, 'Jitka', 3000)$
- D  $zamestnanec = (\{1, 'Jan', 1000\}, \{2, 'Jana', 2000\}, \{3, 'Jitka', 3000\})$
- E  $zamestnanec = \{\{1, 'Jan', 1000\}, \{2, 'Jana', 2000\}, \{3, 'Jitka', 3000\}\}$

**25** Uvažujte následující relace z databáze autopůjčovny: *zákazník*(*rč*, *jméno*, *adresa*), *auto*(*spz*, *typ*) a *půjčeno*(*id*, *rč*, *spz*, *datum\_od*, *datum\_do*, *cena*). Předpokládejte, že v každé relaci je alespoň jeden záznam a žádná relace neobsahuje hodnoty NULL. Atributy *rč* a *spz* v relaci *půjčeno* jsou cizí klíče do *zákazník* a *auto*.

Jaký výsledek vrátí následující SQL dotaz:

```
SELECT jméno, COUNT(id) FROM zákazník NATURAL LEFT OUTER JOIN půjčeno GROUP BY rč, jméno
```

- A vypíše jména všech zákazníků a počet všech někdy vypůjčených aut.
- B vypíše jméno a počet všech aut, které si někdy půjčil nějaký zákazník.
- C vypíše jméno a počet vypůjčených aut každého zákazníka, který si někdy půjčil alespoň jedno auto.
- \*D vypíše jméno každého zákazníka a počet aut, které si někdy případně vypůjčil.
- E vypíše jména a počty všech zákazníků, kteří si někdy v půjčovně půjčili alespoň jedno auto.

## Softwarové inženýrství

**26** Návrh softwarové architektury bývá motivován několika účely. Které z následujících mezi ně patří (tj. bývají uváděny jako podstatné faktory)?

I. Usnadnit znovupoužitelnost (softwarových komponent).

II. Usnadnit komunikaci (mezi vývojářskými týmy a dalšími zúčastněnými).

III. Rozložit složitost systému a podpořit jeho distribuovatelnost.

IV. Usnadnit udržovatelnost systému.

- A Pouze I., II. a III., nikoli IV.
- B Pouze I. a III., nikoli II. a IV.
- \*C Všechny I., II., III., IV.
- D Pouze II. a IV., nikoli I. a III.
- E Pouze III. a IV., nikoli I. a II.



**27** Pro následující aktivity - Test planning (plánování testů), Cost estimating (odhad nákladů na systém), Requirements specification review (zhodnocení specifikace požadavků) - zvolte, do které z následujících fází životního cyklu nejlépe zapadají:

- A Implementace systému
- B Odstavení systému
- C Údržba systému
- D Návrh systému
- \*E Specifikace požadavků

**28** Uvažujme následující příklady vazeb v UML diagramu tříd. Ve kterém případě jsou vazby použity nejhodněji?

- A Auto-Řidič je obecná asociace, Auto-VolvoXC90 je kompozice, Auto-Motor je dědičnost.
- B Auto-Řidič je kompozice, Auto-VolvoXC90 je obecná asociace, Auto-Motor je dědičnost.
- C Auto-Řidič je kompozice, Auto-VolvoXC90 je dědičnost, Auto-Motor je obecná asociace.
- D Auto-Řidič je dědičnost, Auto-VolvoXC90 je obecná asociace, Auto-Motor je kompozice.
- \*E Auto-Řidič je obecná asociace, Auto-VolvoXC90 je dědičnost, Auto-Motor je kompozice.

**29** Které z následujících tvrzení nejlépe odráží změny v diagramu tříd při přechodu od analýzy systému k jeho návrhu (v případě kvalitní analýzy a návrhu)?

- A Při přechodu od analýzy systému k jeho návrhu v diagramu tříd ke změnám nedochází.
- \*B Počet tříd se výrazně zvýší, ale modely jednotlivých tříd se nestanou výrazně složitějšími.
- C Počet tříd ani jejich složitost se nezvýší, ale upřesní se vazby mezi třídami.
- D Počet tříd ani jejich složitost se nezvýší, ale upřesní se argumenty a návratové hodnoty metod ve třídách.
- E Počet tříd zůstane nezměněn, ale modely jednotlivých tříd se stanou výrazně složitějšími.

**30** Jako softwarový inženýr zvažujete, zda nově vznikající systém cílit k nasazení na jediný server nebo ho navrhnout jako distribuovaný (nasazený na více serverů). Uvažujme následující tři argumenty:

I. Pokud jsou výkonnostní charakteristiky řešení na jednom serveru vyhovující, dám přednost nasazení na jeden server, protože přechod do distribuovaného prostředí může přinést snížení spolehlivosti a bezpečnosti systému.

II. Pokud to je finančně únosné, dám vždy přednost distribuovanému řešení, protože je s ním spojena vyšší výkonnost, spolehlivost a bezpečnost.

III. Distribuované řešení s sebou může nést řadu výhod i nevýhod, které je před rozhodnutím nutné pečlivě zvážit.

Rozhodněte, které z uvedených argumentů I., II. a III. jsou obecně validní (zvolte možnost obsahující právě všechny z nich):

- A I., II. a III.
- B II. a III.
- \*C I. a III.
- D II.
- E III.