Zadání a řešení testu z informatiky a zpráva

o výsledcích přijímacího řízení do magisterského navazujícího studia od jara 2016

Zpráva o výsledcích přijímacího řízení

do magisterského navazujícího studia od jara 2016

Počet podaných přihlášek	150
Počet přihlášených uchazečů	138
Počet uchazečů, kteří splnili podmínky přijetí	71
Počet uchazečů, kteří nesplnili podmínky přijetí	67
Počet uchazečů přijatých ke studiu, bez uvedení počtu uchazečů přijatých ke studiu až na základě výsledku přezkoumání původního rozhodnutí	71
Počet uchazečů přijatých celkem	71
Percentil pro přijetí	11,49

Základní statistické charakteristiky

		Matematika	Informatika	Celkem
Počet otázek		25	30	55
Počet uchazečů, kteří se zúčastnili přijímací zkoušky		87	87	87
Nejlepší možný výsledek		25.00	30.00	55.00
Nejlepší skutečně dosažený výsledek		23.75	27.50	51.25
Průměrný výsledek		14.88	16.48	31.36
Medián		15.00	16.75	32.25
Směrodatná odchylka		4.67	5.23	8.69
	Percentil			
Decilové hranice výsledku *	10	8.25	11.00	19.50
	20	11.75	13.25	26.00
	30	13.25	13.75	29.25
	40	14.25	15.50	30.75
	50	15.00	16.75	32.25
	60	15.75	17.75	33.75
	70	17.25	19.50	35.25
	80	19.00	20.75	37.75
	90	20.50	22.50	41.00

* 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*
		12

Algoritmizace a datové struktury

---

1 Předpokládejme maximovou zleva zarovnanou binární haldu. Do prázdné haldy byly vloženy následující prvky v tomto pořadí: 7, 9, 6, 0, 12, 1, 4. Které z uvedených tvrzení pro haldu po tomto vkládání platí?

A Halda má celkem 4 patra.
B Kořenem haldy je prvek s hodnotou 6.
***C** Součet prvků v prostředním patře haldy je 15.
D V listech jsou zleva doprava tyto prvky: 0, 1, 4, 6.
E V listech jsou zleva doprava tyto prvky: 0, 6, 1, 4.

---

2 Které z následujících tvrzení je pravdivé při prohledávání orientovaného grafu?

A Pokud je při prohledávání do šířky vrchol u poprvé navštíven dříve než vrchol v, pak v grafu z u neexistuje cesta do v.
B Pokud je při prohledávání do hloubky vrchol u poprvé navštíven dříve než vrchol v, pak v grafu z u neexistuje cesta do v.
***C** Ani jedno z uvedených tvrzení není pravdivé.
D Pokud je při prohledávání do šířky vrchol u poprvé navštíven dříve než vrchol v, pak v grafu z u existuje cesta do v.
E Pokud je při prohledávání do hloubky vrchol u poprvé navštíven dříve než vrchol v, pak v grafu z u existuje cesta do v.

---

3 Mějme oboustranně spojovaný seznam prvků (doubly linked list), jehož délka je n.
Které z uvedených tvrzení platí?

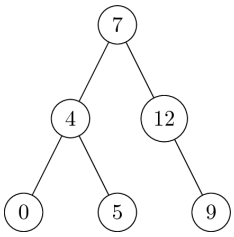
A Časová složitost přístupu na pozici zadanou indexem je $O(1)$.
B Časová složitost přístupu na pozici zadanou indexem je $O(\log n)$, pokud je seznam seřazený.
C Časová složitost nalezení zadaného prvku je $O(\log n)$, pokud je seznam seřazený, jinak je $O(n)$.
D K jednotlivým položkám seznamu se přistupuje podle jmenného klíče.
***E** Časová složitost vložení nového prvku za již nalezený prvek je $O(1)$.

---

4 Které z následujících tvrzení je pravdivé?

A Funkce logaritmus roste asymptoticky stejně rychle jako funkce druhá odmocnina.
B Datové struktury fronta i zásobník pracují na principu FIFO (first in, first out), rozdíl mezi nimi je v tom, že fronta podporuje priority vkládaných prvků.
C To, že je algoritmus parciálně korektní, zaručuje, že na všech vstupech skončí.
D Časová složitost vkládání prvků do hašovací tabulky je vždy $O(\log(n))$.
***E** Mezi binární vyhledávací stromy patří AVL stromy a červeno-černé stromy.

5



Uvažujte vkládání do binárního vyhledávacího stromu. Která z následujících sekvencí hodnot mohla vést k vytvoření binárního stromu uvedeného výše za předpokladu, že strom byl na počátku prázdný a během vkládání nedošlo k vyvažování stromu?

A 7, 12, 4, 5, 9, 0
B 7, 4, 12, 5, 9, 0
***C** Žádná sekvence vkládání do binárního vyhledávacího stromu nevytvoří výše uvedený binární strom.
D 0, 4, 5, 7, 9, 12
E 7, 0, 12, 4, 5, 9

Počítačové systémy

6 Krátkodobý plánovač v operačním systému:

A rozhoduje, kdy který program na disku spustit
B rozhoduje, kdy který spuštěný proces odložit na disk
C vybírá z odložených procesů jeden pro běh na procesoru
***D** vybírá z připravených procesů jeden pro běh na procesoru
E vybírá z čekajících/blokovaných procesů jeden pro běh na procesoru

7 Nechť v časovém okamžiku 0 vznikl požadavek na proces P1, v časovém okamžiku 3 vznikl požadavek na proces P2 a v časovém okamžiku 4 vznikl požadavek na proces P3. Proces P1 potřebuje pro svůj běh 10 časových jednotek CPU, proces P2 potřebuje 2 jednotky CPU a proces P3 potřebuje 4 jednotky CPU. Pokud máme k dispozici jeden procesor a plánování CPU je prováděno algoritmem SJF (shortest job first) s předbíháním, jaký proces bude mít k dispozici procesor v časovém okamžiku 6 jednotek CPU (od okamžiku 0)?

A P2
B P1
***C** P3
D P1 a zároveň P2
E žádný z těchto procesů

8 Tzv. preemptivní plánování se provádí v situaci, kdy nějaké vlákno/proces:

***A** přechází ze stavu běžící do stavu připravený
B přechází ze stavu běžící do stavu čekající/blokovaný
C přechází ze stavu ukončený do stavu připravený
D přechází ze stavu připravený do stavu odložený
E končí

9 Které číslo v desítkové soustavě je ekvivalentem čísla vyjádřeného v šestnáctkové (hexadecimální) soustavě jako A0B?

***A** 2571
B 2661
C 101000001011
D 2561
E 2651

10 V osmibitové reprezentaci se ve dvojkovém doplňkovém kódu dekadické číslo -88 zapíše jako:

A 11010111
B 01010111
C 11011000
***D** 10101000
E 01011000

Programování

---

11 Které z následujících tvrzení **neplatí**?

***A** Volání hodnotou a volání jménem jsou dva různé názvy pro stejný druh volání funkcí.
B Při líné vyhodnocovací strategii ve funkcionálním programování se parametry vyhodnocují až ve chvíli, kdy je potřebné znát jejich hodnotu.
C Při volání referencí se změna parametru uvnitř funkce projeví i navenek.
D Principem výpočtu při logickém programování je logická dedukce.
E V čistě funkcionálních programovacích jazycích nemají funkce žádné vedlejší efekty.

---

12 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 haldě. Po opuštění funkce jsou automaticky dealokovány.
II. V jazycích s garbage collectorem není potřeba explicitně dealokovat dynamickou paměť.
III. Dynamická alokace paměti se typicky provádí pomocí operátoru new.

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

---

13 Rozhodněte, které z uvedených tvrzení je v běžných OOP jazycích (C++, Java, C#) obecně platné:

A Pokud třída B dědí od třídy A, pak má přístup ke všem atributům třídy A, i těm soukromým (private).
***B** Při pozdní vazbě (volání virtuální metody) se o tom, která metoda se přesně zavolá, rozhoduje až za běhu.
C Pojmy třída a objekt jsou totožné, rozlišují se pouze z historických důvodů.
D Pokud je metoda třídy virtuální, její implementaci nelze v potomcích této třídy změnit.
E Rozdíl mezi třídami a objekty je ten, že objekty obsahují pouze atributy, zatímco třídy mohou kromě atributů obsahovat i metody.

---

14 Je dán programový zápis:

```
// PSEUDOKÓD 1
sum = 0
for i = 1 to n {
        sum = <VÝRAZ>
}
print sum

// PSEUDOKÓD 2
sum = 0
i = <INICIÁLNÍ HODNOTA>
while (<PODMÍNKA>) {
        sum = sum + i
        i = i + 1
}
print sum
```

Pro výše uvedené pseudokódy platí, že všechny proměnné jsou typu `integer` (celé číslo) a `n >= 1`. Která z uvedených možností dosazení za části `<VÝRAZ>`, `<INICIÁLNÍ HODNOTA>` a `<PODMÍNKA>` musí být použita, aby byl výstup těchto dvou pseudokódů různý?

A `<VÝRAZ> = sum + i - 1; <INICIÁLNÍ HODNOTA> = 0; <PODMÍNKA> = i < n`
***B** `<VÝRAZ> = sum + i - 1; <INICIÁLNÍ HODNOTA> = 0; <PODMÍNKA> = i <= n`
C `<VÝRAZ> = 3 * n; <INICIÁLNÍ HODNOTA> = n - 1; <PODMÍNKA> = i <= n + 1`
D `<VÝRAZ> = sum + 1; <INICIÁLNÍ HODNOTA> = n; <PODMÍNKA> = i <= n`
E `<VÝRAZ> = sum + i; <INICIÁLNÍ HODNOTA> = 1; <PODMÍNKA> = i <= n`

---

15 Je dán programový zápis:

```
function foo(value integer a, reference integer b)
begin
        integer c
        c = a
        a = b
        b = c
        return c
end

program main()
begin
        integer a = 3
        integer b = 4
        integer c = 5
        a = foo(a, b)
        c = foo(b, c)
        print "a = ", a, ", b = ", b, ", c = ", c
end
```

Předpokládejte, že argument `a` funkce `foo` je volaný hodnotou a argument `b` referencí. Jaký bude výstup programu?

A *a = 4, b = 5, c = 3*
B *a = 3, b = 4, c = 5*
***C** *a = 3, b = 3, c = 3*
D Uvedený kód bude cyklit a nezastaví.
E *a = 3, b = 5, c = 3*

Počítačové sítě

16 Digitální data se pro vysílání reálným přenosovým médiem mohou kódovat samoopravnými kódy. Přijímač v takovém případě může reagovat na chyby vzniklé během přenosu tak, že:

A detekuje některé chyby a tyto opraví až po získání správných dat od vysílače
B detekuje a opraví všechny chyby
***C** některé chyby detekuje, některé chyby opraví a některé chyby nezjistí
D detekuje všechny chyby a z těchto většinu opraví
E detekuje všechny chyby a tyto opraví až po získání správných dat od vysílače

---

17 Řízení zahlcení (Congestion Control) je součástí transportního protokolu TCP. Jedná se o:

A zastaralý již nevyužívaný mechanismus
***B** ochranu sítě (vnitřních uzlů) v případě, že rychlost příchodu paketů je vyšší než rychlost jejich zpracování nebo rychlost zpracování paketů je nižší než rychlost jejich výstupu
C ochranu proti SPAMům u protokolu POP3
D ochranu proti zahlcení vysílajícího koncového uzlu
E ochranu proti zahlcení přijímajícího koncového uzlu

---

18 IPv6 adresy jsou považovány za finální řešení nedostatku IP adres. Pro jejich délku se s nimi špatně pracuje, proto byla zavedena pravidla pro jejich zkracování. Rozviňte IPv6 adresu FEDC:98::7654::2922 do nezkráceného tvaru.

A FEDC:0098::::7654::2922:FFFF
B FEDC:0098:000:000:7654:0000:0000:2922
***C** nelze rozvinout
D FEDC:0098:0000:7654:0000:2922
E FEDC:98:000:000:7654:0000:0000:2922

---

19 Elektromagnetický signál vysílaný v Brně se kabelem rozšíří do Prahy řádově:

A v jednotkách mikrosekund
B v jednotkách picosekund
***C** v jednotkách milisekund
D v jednotkách sekund
E v jednotkách nanosekund

---

20 Simple Mail Transfer Protocol (SMTP) je standardní mechanismus pro posílání elektronické pošty (electronic mail, email) v Internetu. Struktura SMTP emailové zprávy se skládá:

A z obálky obsahující předmět zprávy a adresu příjemce
***B** z obálky obsahující adresu odesílatele, adresu příjemce a případně další informace a z vlastní zprávy obsahující hlavičky zprávy a tělo zprávy
C z obálky obsahující adresu odesílatele a z vlastní zprávy obsahující údaje o odesílateli, příjemci a předmětu zprávy
D z hlaviček definujících příjemce a předmět zprávy
E z hlaviček specifikujících URL, odkud je možné získat tělo zprávy

---

Databázové systémy

---

21 Vyberte správnou posloupnost operací, které relační databáze provádí při zpracování dotazu v jazyce SQL (SELECT):

A syntaktická kontrola SQL, sémantická kontrola SQL, výběr optimálního plánu zpracování dotazu v SQL spouštěním různých variant dotazu na náhodně zvolených relacích, vyhodnocení vybraného plánu na reálných datech, přenos dat zpět k zadavateli dotazu

B pouze sémantická kontrola SQL, výběr optimálního plánu zpracování dotazu v SQL s pomocí statistik o datech v tabulkách, vyhodnocení vybraného plánu na reálných datech, přenos dat zpět k zadavateli dotazu

C syntaktická a sémantická kontrola SQL, překlad dotazu z SQL do relační algebry, vyhodnocení výrazu relační algebry na reálných datech, přenos dat zpět k zadavateli dotazu

D syntaktická a sémantická kontrola SQL, překlad dotazu z SQL do relační algebry, výběr optimálního plánu zpracování výrazu relační algebry spouštěním různých variant dotazu na automaticky vybrané podmnožině dat z relací, vyhodnocení vybraného plánu na reálných datech, přenos dat zpět k zadavateli dotazu

***E** syntaktická a sémantická kontrola SQL, překlad dotazu z SQL do relační algebry, výběr optimálního plánu zpracování výrazu relační algebry s pomocí statistik o datech v tabulkách, vyhodnocení vybraného plánu na reálných datech, přenos dat zpět k zadavateli dotazu

---

22 Mějme relace:

r

A	**B**
a	1
a	2
b	1

s

B
1
2

Kolik řádků bude mít výsledek po vyhodnocení výrazu $r \bowtie s$ (přirozené spojení relací r a s):

A 5

B 6

C 0

D 2

***E** 3

---

23 Máme relaci *kino(nazev_pobocky, cislo_salu, datum, cas, nazev_filmu, jazyk, reziser, cena, kapacita)* a následující množinu funkčních závislostí:
nazev_pobocky, cislo_salu, datum, cas → nazev_filmu, jazyk
nazev_pobocky, datum, cas, nazev_filmu → cislo_salu
nazev_filmu → reziser
nazev_pobocky, nazev_filmu → cena
nazev_pobocky, cislo_salu → kapacita
Co můžeme tvrdit o množině atributů *{nazev_pobocky, cislo_salu, datum, reziser, cena, kapacita}*?

A Je to superklíč, ale není to kandidátní klíč.

B Je to kandidátní klíč, ale není to superklíč.

C Je to jediný primární klíč.

D Je to superklíč i kandidátní klíč.

***E** Není to ani kandidátní klíč, ani superklíč.

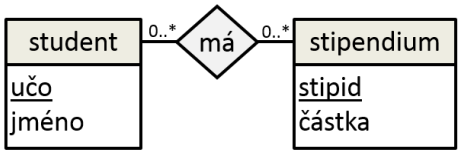
---

24 Uvažujte následující relace z databáze autopůjčovny: *zakaznik(rc, jmeno, adresa)*, *auto(spz, typ, porizovaci_cena)* a *pujceno(id, rc, spz, datum_od, datum_do, cena)*. Předpokládejte, že v relaci *pujceno* má každý záznam vždy vyplněné všechny atributy.
Z následujících SQL dotazů vyberte ten, který vrátí spz aut, jejichž provoz už zaplatil jejich pořizovací cenu:

A SELECT auto.spz FROM pujceno NATURAL INNER JOIN auto WHERE porizovaci_cena <= cena;
B SELECT auto.spz FROM auto, pujceno WHERE auto.spz = pujceno.spz AND porizovaci_cena <= SUM(cena) GROUP BY auto.spz, auto.porizovaci_cena;
C SELECT auto.spz FROM pujceno, auto WHERE porizovaci_cena <= SUM(cena) GROUP BY auto.spz, auto.porizovaci_cena;
D SELECT auto.spz FROM auto, pujceno WHERE auto.spz = pujceno.spz AND porizovaci_cena <= cena;
***E** SELECT auto.spz FROM pujceno NATURAL INNER JOIN auto GROUP BY spz, porizovaci_cena HAVING porizovaci_cena - SUM(cena) <= 0;

---

25 Jestliže převedeme následující E-R diagram na relační model,



bude výsledný relační model obsahovat právě relace (podtržením jsou označeny primární klíče relací):

A student(učo, jméno), stipendium(stipid, částka), má(učo, stipid)
B student(učo, jméno), stipendium(stipid, částka), má(učo, stipid)
C student(učo, jméno), stipendium(stipid, částka)
D student(učo, jméno), stipendium(stipid, částka), má(učo, stipid)
***E** student(učo, jméno), stipendium(stipid, částka), má(učo, stipid)

Softwarové inženýrství

---

26 Uvažujme následující techniky použitelné při vývoji software:
I. Paralelizace probíhajících výpočtů.
II. Optimalizace rozložení zátěže na dostupné výpočetní uzly.
III. Odstranění redundantních výpočtů (opakování stejného výpočtu).
Na který z následujících nefunkčních atributů bude mít kombinace uvedených technik nejsilnější pozitivní dopad?

A testovatelnost (testability)
B spolehlivost (reliability)
***C** výkonnost (performance)
D bezpečnost (security)
E udržovatelnost (maintainability)

---

27 Který z následujících bodů **nepatří** mezi zásady agilního vývoje?

A Upřednostnění jednotlivců a interakcí před procesy a nástroji.
B Upřednostnění spolupráce se zákazníkem před vyjednáváním o smlouvě.
C Upřednostnění reagování na změny před dodržováním plánu.
D Upřednostnění fungujícího software před vyčerpávající dokumentací.
***E** Upřednostnění dodávky funkčního celku před dodáním po funkčních částech.

---

28 Jako softwarový inženýr vedete vývoj informačního systému pro nově vznikající společnost, která klade důraz na cenu, ale nemá dosud zcela vyjasněné požadavky na systém. Dáte přednost vodopádovému nebo inkrementálnímu modelu vývoje a proč?

***A** Zvolím inkrementální model, protože tak dám společnosti možnost začít brzy používat první verze systému, které mu pomohou ve vyjasnění požadavků.
B Zvolím vodopádový model, protože povede k nižší ceně vývoje.
C Zvolím vodopádový model, protože je historickým nástupcem inkrementálního modelu, který už dnes není vhodné používat.
D Zvolím inkrementální model, protože je historickým nástupcem vodopádového modelu, který už dnes není vhodné používat.
E Zvolím vodopádový model, protože je zástupcem agilních metodik, které nabízí prostředky pro rychlý vývoj systémů s nejasnými požadavky.

---

29 Uvažujme následující techniky: testování jednotek (unit testing), akceptační testování (acceptance testing), uživatelské testování (user testing), zátěžové testování (stress testing), regresní testování (regression testing). Které z nich patří mezi techniky verifikace a validace software?

A Žádná z uvedených.
***B** Všechny uvedené.
C Všechny vyjma regresního testování (regression testing).
D Všechny vyjma uživatelského testování (user testing).
E Všechny vyjma zátěžového testování (stress testing).

---

30 Jako softwarový inženýr máte pro každé z uvedených paradigmat zvolit vhodný programovací jazyk. Která z následujících možností přiřazuje jazyky správně?

A Procedurální (Prolog), objektově orientované (Java), funkcionální (C), logické (Haskell).
B Procedurální (C), objektově orientované (Java), funkcionální (Prolog), logické (Haskell).
C Procedurální (Java), objektově orientované (C), funkcionální (Prolog), logické (Haskell).
D Procedurální (Java), objektově orientované (C), funkcionální (Haskell), logické (Prolog).
***E** Procedurální (C), objektově orientované (Java), funkcionální (Haskell), logické (Prolog).
