

Přijímací zkouška - Informatika

Jméno a příjmení - pište do okénka	Číslo přihlášky	Číslo zadání
		1

Algoritmizace a datové struktury

- 1** Která z uvedených vlastností je platná pro datovou strukturu pole o n prvcích (hodnoty jsou v paměti uloženy přímo za sebou):
- A k jednotlivým položkám se přistupuje na základě jmenného klíče
 - *B v případě, že neznáme pozici (index) hledané položky, tak je časová složitost přístupu k ní $O(n)$
 - C struktura pole je výhodná pro použití v situacích, kdy potřebujeme často vkládat dodatečný prvek mezi existující prvky na libovolné pozici
 - D pole má ve srovnání se strukturou oboustranně spojovaného seznamu prvků typicky větší paměťové požadavky
 - E časová složitost přístupu na pozici zadanou indexem je $O(n)$
-
- 2** Předpokládejte, že máte jako vstup dvě pole reálných čísel o n prvcích a jako výstup máte poskytnout jediné pole, které bude obsahovat všechny položky z těchto dvou vstupních polí a které bude seřazené vzestupně. Vstupní pole jsou již seřazena vzestupně a mohou existovat prvky, které se vyskytují v obou polích. Uvažujme algoritmus vytvářející takovéto spojení. Které z uvedených tvrzení platí?
- A Nejnižší časovou složitost dosáhneme kopírováním obou polí do jediného a aplikací řadícího algoritmu quicksort s výslednou složitostí $O(\log(n)*n)$.
 - B Nejnižší časovou složitost dosáhneme kopírováním obou polí do jediného a aplikací řadícího algoritmu bubblesort s výslednou složitostí $O(n*n)$.
 - *C Nejnižší časovou složitost dosáhneme postupným kopírováním obsahu vstupních polí do výstupního tak, aby bylo pořadí zachováno vzestupné uspořádání s výslednou složitostí $O(n)$.
 - D Nejnižší časovou složitost dosáhneme kopírováním obou polí do jediného výstupního pole bez nutnosti dalšího řazení s výslednou složitostí $O(1)$.
 - E Nejnižší časovou složitost dosáhneme převedením vstupních polí do binárního vyhledávacího stromu a následným uložením listů tohoto stromu zpět do výstupního pole s celkovou složitostí $O(\log(n)*n)$.
-
- 3** Předpokládejme existenci oboustranně spojovaného seznamu prvků (double-linked list), u kterého máme uložený ukazatel na poslední prvek. Délka seznamu je n . Které z uvedených tvrzení platí?
- A Časová složitost vložení nového prvku za poslední prvek je $O(n)$.
 - B Časová složitost získání prvního prvku v seznamu je $O(1)$.
 - *C Časová složitost nalezení zadaného prvku je $O(n)$.
 - D Časová složitost vložení nového prvku za již nalezený existující prvek je $O(n)$.
 - E Paměťová režie pro realizaci spojovaného seznamu je stejná jako pro pole prvků uložených v paměti kontinuálně za sebou.
-
- 4** Pro datovou strukturu známou jako vyvážený binární strom platí:
- A vyvážený binární strom není nutné během vkládání nových uzlů vyvažovat
 - B výška vyváženého binárního stromu je vzhledem k počtu uzlů v nejhorším případě lineární
 - C má-li n uzlů, pak obsahuje právě n hran
 - D operace vyhledávání v binárním stromu s n uzly má časovou složitost $O(\log(\log(n)))$
 - *E operace odstranění uzlu z vyváženého binárního stromu s n uzly má časovou složitost $O(\log(n))$
-

- 5** Předpokládejte, že máme N síťových prvků, některé z nich jsou propojeny pomocí kabelových rozvodů. Úkolem je zjistit, zda je možné ustanovit datový spoj (přes jeden nebo více fyzických kabelů) z každého síťového prvku X do libovolného jiného síťového prvku Y . Která z uvedených možností je optimálním řešením?
- A** Hledání nejdelšího tunelu v grafu síťových prvků. Pokud má tunel alespoň 2 uzly, pak se nelze z některého X dostat do některého Y .
 - B** Procházení grafu síťových prvků do hloubky. Pokud je objeven v grafu cyklus, tak se nelze z některého X dostat do některého Y .
 - *C** Hledání minimální kostry grafu síťových prvků. Pokud taková kostra neexistuje, tak se z některého X nelze dostat do některého Y .
 - D** Vyzkoušení všech možných cest v grafu síťových prvků hrubou silou. Pokud se některou cestu nalézt nepodaří, tak se z některého X nelze dostat do některého Y .
 - E** Hledání nejkratšího cyklu v grafu síťových prvků. Pokud je objeven v grafu cyklus, tak se z některého X nelze dostat do některého Y .
-

Programování

6

```
a = 3;
b = 0;
c = 0;
while (b < 3) {
    while (a > b) {
        c = c + 1;
        b = b + 1;
    }
    a = a - 1;
    b = b + 1;
}
```

Pro obsah proměnných a , b , c po konci uvedeného kódu bude platit:

- *A** $a = 2, b = 4, c = 3$
 - B** $a = 3, b = 0, c = 0$
 - C** $a = 3, b = 4, c = 3$
 - D** $a = 2, b = 5, c = 3$
 - E** uvedený kód bude cyklit a nezastaví
-

```
7 int offset = 0;
  int dataset[10];
  void put(int value) {
    if (offset < 10) {
      dataset[offset] = value;
      offset = offset + 1;
    }
  }
  int get() {
    if (offset > 0) {
      if (offset < 11) {
        offset = offset - 1;
        return dataset[offset];
      } else return -1;
    } else return -1;
  }
}
```

Výše uvedený kód poskytuje dvě základní funkce pro práci s jednou z často používaných datových struktur. Proměnné *offset* a *dataset* jsou globální proměnné. Datový typ *int* značí celá čísla se znaménkem, *dataset[offset]* vrací hodnotu v poli *dataset* na pozici *offset+1* (indexování od nuly). Rozhodněte, která z uvedených možností je správná pro výše uvedený kód:

- A funkce implementují datovou strukturu FIFO pro neomezený počet prvků
 - B funkce implementují datovou strukturu LIFO pro neomezený počet prvků
 - C funkce implementují datovou strukturu zřetězený seznam (list) pro neomezený počet prvků
 - D funkce implementují datovou strukturu FIFO pro omezený počet prvků
 - *E funkce implementují datovou strukturu LIFO pro omezený počet prvků
-

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

- A objekt je instancí třídy, třída je instancí objektu
 - B pro daný předlohový objekt může existovat více instancí, které označujeme pojmem třída
 - C z jedné třídy může být vytvořen vždy právě jeden objekt
 - D třída definuje hlavičky metod, objekt jejich implementaci
 - *E třída obsahuje deklaraci nabízených metod a může obsahovat i jejich implementaci
-

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

- *A pokud je metoda třídy virtuální, lze její implementaci změnit v potomcích této třídy
 - B pokud je metoda třídy virtuální, nelze změnit její implementaci v potomcích této třídy
 - C režie při zavolání virtuální metody je typicky nižší než při zavolání metody volané včasnou vazbou (nevirtuální)
 - D metodu lze označit jako virtuální pouze v rozhraní (Java - interface, C++ - abstraktní třída)
 - E pokud metoda není virtuální v dané třídě *X*, tak ji v potomcích třídy *X* nelze na virtuální předeclarovat
-

- 10** Předpokládejme, že třída *A* má metody *publicFoo()*, *protectedFoo()* a *privateFoo()* s právy přístupu pro *publicFoo()* jako *public*, *protectedFoo()* jako *protected* a *privateFoo()* jako *private*. Rozhodněte, které z uvedených tvrzení je v běžných OOP jazycích (C++, Java, C#) platné:
- A Objekt *X* může volat všechny metody třídy *A* v případě, že je instancí potomka třídy *A*.
 - B Objekt *X* vždy může volat pouze metody *publicFoo()* a *protectedFoo()* a nemůže volat *privateFoo()*, nezávisle na tom, zda je instancí potomka třídy *A*.
 - C Objekt *X* vždy může volat pouze metodu *publicFoo()* a nemůže volat *protectedFoo()* a *privateFoo()*, nezávisle na tom, zda je instancí potomka třídy *A*.
 - D Objekt *X* může volat pouze metodu *publicFoo()* v případě, že se nejedná o instanci potomka třídy *A*. Nemůže volat *protectedFoo()* a *privateFoo()*, nezávisle na tom, zda je instancí potomka třídy *A*.
 - *E Objekt *X* vždy může volat pouze metodu *publicFoo()*. Metodu *protectedFoo()* může volat v případě, že je instancí přímého potomka třídy *A*. Metodu *privateFoo()* může volat pouze v případě, že je instancí třídy *A*.

Databáze

- 11** B-strom je:
- A zkratka pro binární vyhledávací strom
 - B rozšíření A-stromu
 - C strom, kde každý uzel má vždy právě dva následníky
 - *D *n*-ární vyvážený strom
 - E strom, kde každý uzel je zároveň i kořenem
- 12** Předpokládejme, že relační databáze obsahuje tabulku *ctenari* s klíčem (*id_ctenar*), tabulku *knihy* s klíčem (*id_kniha*) a tabulku *pujceno* s klíčem (*co*, *komu*). Jaký výsledek vrátí následující SQL dotaz:
- ```
SELECT ctenari.id_ctenar, count(*)
FROM ctenari, knihy, pujceno
WHERE pujceno.co=knihy.id_kniha and pujceno.komu=ctenari.id_ctenar
and knihy.nazev LIKE '%SQL%'
GROUP BY ctenari.id_ctenar;
```
- A seznam identifikátorů všech knih v knihovně a počet kolikrát již byly vypůjčeny
  - B seznam identifikátorů čtenářů, kteří si nikdy nepůjčili žádnou knížku
  - \*C seznam identifikátorů čtenářů a počet jimi vypůjčených knih, které mají v názvu řetězec SQL
  - D počet čtenářů knihovny
  - E seznam identifikátorů čtenářů a názvy jimi vypůjčených knih, které mají v názvu řetězec SQL

- 13** Uzávěr množiny funkčních závislostí lze nalézt pomocí:
- A Boyce-Coddových axiomů
  - \*B Armstrongových axiomů
  - C první normální formy
  - D Chomského normální formy
  - E Hornových axiomů

- 14** Vyberte **nepravdivé** tvrzení ohledně normalizace relačního schématu:
- A normalizace se provádí, aby schéma odpovídalo normálním formám
  - \*B normalizace může znamenat ztrátu dat
  - C při normalizaci může být nutné rozdělit jednu tabulku do dvou tabulek
  - D častým problémem, který při normalizaci řešíme, je odstranění redundance dat
  - E algoritmus normalizace bere v úvahu i funkční závislosti
-

**15** Vyberte **nepravdivé** tvrzení o kandidátním klíči (KK):

- A Pro jednu entitní množinu může existovat více KK.
- B KK je klíč.
- C KK je super klíč.
- D Primární klíč je zvolený KK.
- \*E KK je libovolná podmnožina atributů.

---

## Počítačové sítě

---

**16** Přenos jednobitové informace mezi stacionárním satelitem a pozemní bránou trvá:

- A 270 sekund
- B 270 picosekund
- C 270 nanosekund
- \*D 270 milisekund
- E 270 mikrosekund

**17** Referenční model ISO OSI předepisuje uspořádat funkčnosti distribuovaných systémů do hierarchicky uspořádaných vrstev:

- A fyzické, spojové, síťové, transportní, aplikační
- B fyzické, spojové, síťové, transportní, relační, aplikační
- C fyzické, spojové, síťové, transportní, relační, bezpečnostní, prezentační, aplikační
- \*D fyzické, spojové, síťové, transportní, relační, prezentační, aplikační
- E fyzické, spojové, síťové, transportní, transakční, bezpečnostní, prezentační, aplikační

**18** Protokol http implicitně používá port číslo:

- A 22
- B 25
- \*C 80
- D 88
- E 110

**19** Spojovaný komunikační protokol (connection oriented) zaručuje doručení všech vyslaných dat v originálním pořadí, ve kterém byla data vysílána. Vyberte platné tvrzení:

- A spojovaný komunikační protokol lze použít pouze mezi uzly bezprostředně spojenými přenosovým médiem
- B spojovaný komunikační protokol lze použít pouze mezi uzly v jedné síťové doméně
- C spojovaný komunikační protokol lze použít pouze mezi uzly splňujícími omezení daná modelem ISO OSI
- \*D použití spojovaných protokolů není nutnou podmínkou pro dosažení spolehlivé komunikace v prostředí s poruchami
- E použití spojovaných protokolů je nutnou podmínkou pro dosažení spolehlivé komunikace v prostředí s poruchami

**20** Místy poskytování transportních služeb v síti Internet na bázi protokolů TCP/IP jsou:

- A určené systémové procesy (demons)
- \*B numericky identifikovatelné datové schránky (ports)
- C rozhraní ovladačů komunikačních spojů v jádru operačního systému (drivers)
- D rozhraní řadičů komunikačních spojů (control units)
- E MAC adresy

---

## Počítačové systémy

---

- 21** Které číslo ve dvojkové soustavě je ekvivalentem čísla vyjádřeného v šestnáctkové (hexadecimální) soustavě jako 29AF?
- A 0001 1001 1010 1111
  - \*B 0010 1001 1010 1111
  - C 1111 1010 1001 0010
  - D 1001 0010 1010 1111
  - E 10671
- 
- 22** Vyberte **nepravdivé** tvrzení o reprezentaci reálných čísel na počítačích:
- A reálná čísla bývají v počítači uložena podle standardu IEEE 754
  - B reálná čísla se obvykle reprezentují pomocí znaménka, exponentu a mantisy
  - C u reprezentace reálných čísel můžeme použít odlišné techniky pro reprezentaci čísel v pevné řádové čárce a plovoucí řádové čárce
  - D některé reprezentace reálných čísel podporují hodnotu nekonečno
  - \*E významným problémem všech reprezentací reálných čísel v počítači je tzv. problém boolovské algebry
- 
- 23** Vyberte pravdivé tvrzení o přerušení (interrupt) v počítačových systémech:
- A může být vyvoláno symetricky nebo asymetricky
  - B žádné přerušení nelze maskovat
  - C přerušení jsou vždy nedeterministická
  - D v moderních OS si každý proces vytváří svou tabulku přerušení
  - \*E přerušení lze obvykle vyvolat i pomocí speciální instrukce
- 
- 24** Pro synchronizaci a komunikaci mezi procesy se v běžných OS **nepoužívá**:
- A roura (pipe)
  - B semafor
  - \*C most
  - D mutex
  - E sdílená paměť
- 
- 25** Přerušení „výpadek stránky“ je vyvoláno:
- A při hardwarové chybě paměťového čipu
  - \*B při referenci adresy, která není přítomna ve fyzickém adresovém prostoru
  - C při každém přístupu do virtuální paměti
  - D při kontrole přístupových práv k souborům
  - E při chybě komunikace s V/V zařízením
-