

Přijímací zkouška - informatika

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

Algoritmizace a datové struktury

1 Pro datovou strukturu binární vyhledávací strom **platí**:

- A obsahuje nejvýše jeden cyklus, kořen, uzly a listy;
 - B má-li n uzlů, pak obsahuje právě n hran;
 - *C výška stromu je vzhledem k počtu uzlů nejhůře lineární;
 - D operace vyhledávání ve stromu s n uzly má časovou složitost $O(\log(\log(n)))$;
 - E výška stromu je vzhledem k počtu uzlů nejhůře kvadratická;
-

2 Hašovací funkce se v databázích používají, protože:

- *A umožňují vyhledávat položky s časovou složitostí $O(1)$;
 - B umožňují uložit velké množství položek s následným vyhledáním s časovou složitostí $O(\log(n))$;
 - C zajišťují automatické odstranění duplicitních položek;
 - D hašovací funkce se v databázích typicky nepoužívají;
 - E poskytují kompresi ukládaných položek;
-

3 Pro datovou strukturu známou jako zásobník **platí**:

- A naposledy vložený prvek bude odebrán jako poslední;
 - B prvek s nejmenší hodnotou je vždy udržován na vrcholu zásobníku;
 - *C prvek vložený jako první bude odebrán jako poslední;
 - D struktura obsahuje vždy alespoň dva prvky (dno a vrchol zásobníku);
 - E prvek s nejmenší hodnotou je vždy udržován na dně zásobníku;
-

4 Předpokládejme, že máme n měst, některá jsou spojena silnicí. Této struktuře budeme říkat graf měst. Úkolem je zjistit, zda existuje cesta z každého města X do každého města Y . Která z uvedených možností **je optimálním řešením**?

- A Procházení grafu měst do šířky. Pokud je objeven v grafu cyklus, tak se nelze z některého X dostat do některého Y .
 - B Vyzkoušení všech možných cest v grafu měst hrubou silou. Pokud se některou cestu nalézt nepodaří, tak se z některého X nelze dostat do některého Y .
 - *C Hledání minimální kostry grafu měst. Pokud taková kostra neexistuje, tak se z některého X nelze dostat do některého Y .
 - D Procházení grafu měst do hloubky. Pokud je hloubka větší než $O(\log(n))$, tak se nelze z každého X dostat do každého Y .
 - E Hledání nejkratšího cyklu v grafu měst. Pokud je objeven v grafu cyklus, tak se z každého X nelze dostat do každého Y .
-

5 Která z uvedených vlastností **platí** pro datovou strukturu pole (hodnoty jsou v paměti uloženy přímo za sebou)?

- A K jednotlivým položkám se přistupuje na základě jmenného klíče.
 - B Pole obsahuje vždy dvojici klíč a sekundární klíč.
 - C Pole obsahuje vždy dvojici klíč a hodnota.
 - D V případě, že neznáme pozici (index) hledané položky, tak je časová složitost přístupu k ní $O(1)$.
 - *E Časová složitost přístupu na pozici zadanou indexem je $O(1)$.
-

Programování

- 6** Rozhodněte, které z uvedených tvrzení je v běžných objektově orientovaných programovacích jazycích (C++, Java, C#) obecně **platné**:
- A pro danou třídu může existovat pouze jediná její instance;
 - B pojem třída se v objektově orientovaném programování nepoužívá, správné označení je objekt;
 - C třída je instancí objektu;
 - *D z jedné třídy může být vytvořeno několik objektů;
 - E třída definuje hlavičky metod, objekt jejich implementaci;
-
- 7** Rozhodněte, které z uvedených tvrzení je v běžných objektově orientovaných programovacích jazycích (C++, Java, C#) **platné**:
- A zapouzdření je automaticky zajišťováno běhovým prostředím na úrovni ochrany paměti bez nutnosti specifikovat přístupová práva;
 - B prostřednictvím přístupových práv lze řídit přístup k atributům třídy, ale ne k jejím metodám;
 - C zapouzdření pomáhá zjednodušit hlavičky metod skrytím všech jejich argumentů;
 - D zapouzdření umožňuje skrýt atributy, neumožňuje ale skrýt metody třídy;
 - *E pomocí přístupových práv lze řídit přístup k atributům i metodám třídy - skrytím detailů vnitřního stavu objektu se zvyšuje robustnost implementace;
-
- 8** Rozhodněte, které z uvedených tvrzení je v běžných objektově orientovaných programovacích jazycích (C++, Java, C#) **platné**:
- A standardní knihovna neobsahuje žádné třídy pro výjimku - výjimky jsou vždy uživatelsky definované třídy;
 - B pokud není výjimka obsloužena v aktuální funkci, tak se výjimka zahazuje a pokračuje se další instrukcí ve volající funkci;
 - *C pokud není výjimka obsloužena v aktuální funkci, tak se propaguje o úroveň výš do volající funkce;
 - D kód, který může vyvolat výjimku, která je zachytávána, je obalen klauzulí `catch {}`;
 - E po obsloužení výjimky se pokračuje v kódu na řádku následujícím po řádku, ve kterém došlo k vyvolání výjimky;
-
- 9** Předpokládejme, že proměnná pole je souvislé pole o 10 prvcích obsahující různé hodnoty v náhodném pořadí. Indexování pole je od 0, tj. první prvek je na pozici `pole[0]`, druhý na `pole[1]`, ... poslední desátý prvek na pozici `pole[9]`. Uvažujme následující kód programu.
- ```
index1 = 0;
while (index1 < 10) {
 index2 = index1 + 1;
 while (index2 < 10) {
 if (pole[index1] > pole[index2]) {
 tmp = pole[index1];
 pole[index1] = pole[index2];
 pole[index2] = tmp;
 }
 index2 = index2 + 1;
 }
 index1 = index1 + 1;
}
```
- Rozhodněte, která z uvedených možností je **správná** po provedení výše uvedeného kódu:
- A program bude cyklit a korektně neskončí;
  - B program seřadí položky sestupně;
  - \*C program seřadí položky vzestupně;
  - D program ponechá na stejném místě v poli všechny položky větší nebo rovny číslu 10;
  - E program ponechá položky pole v nezměněném pořadí;
-

**10** Uvažujme následující kód programu.

```
a = 0;
b = 5;
c = 0;
while (a < b) {
 b = 7;
 while (b > a) {
 b = b - 1;
 c = c + 1;
 }
 a = a + 1;
}
```

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

- A  $a = 0, b = 5, c = 0$ ;
- B uvedený kód bude cyklit a nezastaví;
- \*C  $a = 1, b = 0, c = 7$ ;
- D  $a = 2, b = 2, c = 35$ ;
- E  $a = 1, b = 7, c = 0$ ;

---

## Databáze

---

**11** Mezi integritní omezení používané v databázových systémech **nepatří**:

- A uživatelem definované omezení;
- B omezení zakazující nedefinované hodnoty;
- C referenční omezení;
- \*D omezení sekundárního klíče;
- E doménové omezení;

**12** Co je to B-strom?

- A Strom, kde každý uzel je zároveň listem.
- B Zkratka pro binární vyhledávací strom.
- C Graf obsahující maximálně jeden cyklus.
- \*D N-ární vyvážený strom.
- E Strom, kde každý uzel má maximálně dva potomky.

**13** Vyberte **nepravdivé** tvrzení o superklíči (SK):

- A pro jednu entitní množinu může existovat více SK;
- B SK je klíč;
- C SK může obsahovat více než jeden atribut;
- \*D SK je vybraný primární klíč;
- E SK jednoznačně identifikuje jednotlivé entity entitní množiny;

**14** Předpokládejme, že relační databáze obsahuje tabulky `studenti(id, jmeno, rocnik)` a `zapsane_predmety(predmet_id, student_id, hodnoceni)`. Uvažujme následující SQL dotaz:

```
SELECT studenti.id, count(*)
FROM studenti, zapsane_predmety
WHERE studenti.id=zapsane_predmety.student_id AND studenti.rocnik=1
GROUP BY studenti.id;
```

Jaký výsledek vrátí výše uvedený dotaz?

- A Počet studentů prvního ročníku a jejich identifikátory.
  - \*B Identifikátory studentů prvního ročníku a počet jimi zapsaných předmětů.
  - C Počet předmětů, které si zapsal právě jeden student prvního ročníku.
  - D Identifikátory předmětů a počet studentů, kteří si je zapsali.
  - E Identifikátory všech studentů a počet jimi zapsaných předmětů.
-

**15** Vyberte **nepřavdivé** tvrzení o normálních formách:

- A relační schéma  $R$ , které je ve třetí normální formě, je zároveň v první normální formě;
- \*B** relační schéma  $R$ , které je v první normální formě, je zároveň ve třetí normální formě;
- C relační schéma  $R$ , které je v Boyce-Coddově normální formě, je zároveň ve druhé normální formě;
- D relační schéma  $R$ , které je ve druhé normální formě, je zároveň v první normální formě;
- E relační schéma  $R$ , které je v Boyce-Coddově normální formě, je zároveň ve třetí normální formě;

---

## Počítačové sítě

---

**16** K přístupu na konkrétní službu cílového uzlu v počítačové síti Internet **je nutné minimálně znát**:

- \*A** IP adresu a místo poskytování transportních služeb (port);
- B místo poskytování transportních služeb (port);
- C IP adresu;
- D MAC adresu;
- E MAC adresu, IP adresu a místo poskytování transportních služeb (port);

**17** Metody řízení přístupu k přenosovému médium založené na principu detekce nosného signálu třídy CSMA:

- A plně eliminují kolize vysílání pouze v bezdrátových sítích;
- \*B** omezují počet kolizí vysílání;
- C jsou implementací filozofie centrálně řízeného přístupu k přenosovému médium;
- D plně eliminují kolize vysílání v bezdrátových i kabelových sítích;
- E umožňují souběžně vysílat více signálů;

**18** Spojovaný komunikační protokol zaručuje:

- A doručení všech vyslaných dat v originálním pořadí ve stanoveném časovém intervalu;
- B doručení všech vyslaných dat v originálním pořadí v časovém intervalu stanoveném kvalitou spoje;
- \*C** doručení všech vyslaných dat v originálním pořadí;
- D doručení všech vyslaných dat bez záruky zachování originálního pořadí;
- E doručení všech vyslaných dat ve stanoveném časovém intervalu;

**19** Kvalitu přenosu dat mezi aplikačními procesy běžícími v síťovém prostředí řeší podle referenčního modelu ISO OSI protokoly:

- A fyzické vrstvy;
- B aplikační vrstvy;
- C prezentační vrstvy;
- D relační vrstvy;
- \*E** transportní vrstvy;

**20** Směrování datových toků mezi koncovými uzly v rozlehlé síti Internet je řešeno:

- A evolučními směrovacími algoritmy;
- \*B** dynamickými distribuovanými směrovacími algoritmy;
- C pravděpodobnostními směrovacími algoritmy;
- D statickými směrovacími algoritmy;
- E dynamickým centrálním směrovacím algoritmem;

---

## Principy počítačových systémů

---

**21** Zvolte **neexistující** architekturu jádra OS:

- \*A** kruhové jádro;
  - B hybridní jádro;
  - C monolitické jádro;
  - D modulární jádro;
  - E mikrojádro;
-

**22** Vyberte **nepravdivé** tvrzení o přerušení (interrupt) v počítačových systémech:

- A může být asynchronní;
- B některá přerušení lze maskovat;
- \*C přerušení typicky obsluhuje rutina uživatelského procesu;
- D umožňuje reagovat na signál ze zařízení;
- E může být vyvoláno i synchronně;

**23** Které číslo v šestnáctkové (hexadecimální) soustavě je ekvivalent čísla vyjádřeného ve dvojkové soustavě binárním řetězem 1111000101100?

- \*A 1E2C
- B 1E28
- C 2E2C
- D F160
- E 1E2D

**24** Vyberte **pravdivé** tvrzení:

- A Algoritmus LRU (Least Recently Used) vybírá pro odstranění z paměti takový blok, který byl použit jako poslední.
- B K virtuální paměti mohou přistupovat pouze virtuální procesory.
- C Instrukce programu jsou procesorem prováděny vždy ve stejném pořadí, v jakém jsou zapsány.
- \*D Procesor používá zásobník pro ukládání návratových adres při volání podprogramů (např. instrukcí CALL procesorů x86).
- E Problém malého a velkého endianu (little/big endian) není u 64bitových procesorů třeba řešit.

**25** Pro virtuální paměť **neplatí**:

- A pokud dojde k referenci stránky, která není přítomna ve fyzickém adresovém prostoru, bude vyvoláno přerušení výpadku stránky;
- B při malé kapacitě fyzického adresového prostoru může dojít k výprasku (thrashing), který značně snižuje výkon počítačového systému;
- \*C při stránkování na žádost musí aplikační proces explicitně požádat o zavedení odložené stránky do fyzického adresového prostoru;
- D virtuální paměť lze implementovat za pomoci stránkování;
- E virtuální paměť lze implementovat za pomoci segmentace;

*Tato strana je prázdná.*