

Začeto dne	ponedeljek, 11. november 2019, 12:19
Stanje	Zaključeno
Dokončano dne	ponedeljek, 18. november 2019, 00:00
Porabljeni čas	6 dni 11 ure
Točke	7,00/7,00
Ocena	10,00 od možne ocene 10,00 (100%)

Vprašanje **1**

Pravilno

Ocena 1,00 od 1,00

Podatkovni tip SET implementiramo s povezanim seznamom. Naj bo  $n$  moč množice  $S1$  in  $m$  moč množice  $S2$ .  
Določi časovno kompleksnost naslednjih operacij:

vstavljanje elementa v množico $S1$	<div>O(n) ▾</div>	✓
dodajanje v množico $S1$ vseh elementov množice $S2$	<div>O(m*n) ▾</div>	✓
brisanje iz množice $S1$ vseh elementov, ki se nahajajo tudi v množici $S2$	<div>O(m*n) ▾</div>	✓

Vaš odgovor je pravilen.

Pravilen odgovor je: vstavljanje elementa v množico  $S1 \rightarrow O(n)$ , dodajanje v množico  $S1$  vseh elementov množice  $S2 \rightarrow O(m*n)$ , brisanje iz množice  $S1$  vseh elementov, ki se nahajajo tudi v množici  $S2 \rightarrow O(m*n)$

Vprašanje **2**

Pravilno

Ocena 1,00 od 1,00

Kakšna je tesna zgornja meja časovne zahtevnosti naslednjega odseka programske kode (izberi NAJBOLJ POENOSTAVLJEN zapis časovne zahtevnosti)? Upoštevaj, da sta  $n$  in  $m$  parametra algoritma.

```
int f = 1;
for(int i=1 ; i<m ; i=i*4) {
    int l = 5;
    while(l < 5000) {
        l = l^2;
        f *= 2;
    }

    for(int j=1 ; j<n ; j++) {
        for(int k=1 ; k<5*j ; k++) {
            f++;
        }
    }
    f += 10;
}
```

- ☐  $O(m^2 * n)$
- ☐  $O(m^2 + n)$
- ☐  $O(n)$
- ☐  $O(\log(m))$
- ☒  $O(\log(m) * n^2)$  ✓
- ☐  $O(\log(m) * \log(n) * n)$
- ☐  $O(m * (n+m))$
- ☐  $O(\log(m^2 * n))$
- ☐  $O(m * (\log(n) + m))$
- ☐  $O(\log(m) * (\log(n))^2)$
- ☐  $O(\log(m) + n^2)$

Vaš odgovor je pravilen.

Pravilen odgovor je:  $O(\log(m) * n^2)$

Vprašanje **3**

Pravilno

Ocena 1,00 od 1,00

Kaj od naslednjega definira abstraktni podatkovni tip?

Izberite enega ali več odgovorov:

- ☒ množica dovoljenih operacij ✓
- ☐ amortizirana analiza časovne zahtevnosti
- ☐ ocena dejanskega časa izvajanja
- ☐ vmesnik za implementacijo
- ☐ način implementacije vseh operacij
- ☐ časovne zahtevnosti operacij
- ☒ množica možnih vrednosti podatkov ✓
- ☐ podatkovni tip za implementacijo abstraktnega podatkovnega tipa

Vaš odgovor je pravilen.

Pravilni odgovori so: množica dovoljenih operacij, množica možnih vrednosti podatkov

Vprašanje 4

Pravilno

Ocena 1,00 od 1,00

Za dani program so bili izmerjeni naslednji časi izvajanja za različne velikosti vhodnih podatkov:

velikost podatkov:	50	100	150	200
čas:	822	1572	2322	3072

Katera funkcija  $f(n)$  najbolj ustreza časovni zahtevnosti  $T(n) = a \cdot f(n) + c$  tega programa v odvisnosti od velikosti vhodnih podatkov?

- ☐  $n^3$
- ☐  $\log(n)$
- ☐  $n^2$
- ☐  $n^2 \cdot \log(n)$
- ☐  $n \cdot \log(n)$
- ☒  $n$  ✓

Vaš odgovor je pravilen.

Pravilen odgovor je:  $n$

Vprašanje 5

Pravilno

Ocena 1,00 od 1,00

Katere izmed spodnjih trditev v zvezi z rekurzijo so resnične?

Izberite enega ali več odgovorov:

- ☒ s pomočjo sklada je možno vsako rekurzijo prevesti v iteracijo ✓
- ☐ rekurzija je pomnilniško manj zahtevna od iteracije
- ☒ vsaka (pravilna) rekurzivna definicija potrebuje robni pogoj ✓
- ☐ samo repno rekurzijo lahko spremenimo v iteracijo

Vaš odgovor je pravilen.

Pravilni odgovori so: vsaka (pravilna) rekurzivna definicija potrebuje robni pogoj, s pomočjo sklada je možno vsako rekurzijo prevesti v iteracijo

Vprašanje 6

Pravilno

Ocena 1,00 od 1,00

Razvijamo neko aplikacijo in želimo izbrati primerno implementacijo za ADT LIST. Primernost izbire temeljimo na časovni zahtevnosti pogostih operacij.

Možnost A je implementacija s statičnim poljem. Možnost B je implementacija v obliki enosmernega seznama s kazalci, pri čemer imamo neposreden dostop do prvega in zadnjega elementa seznama (kazalca first in last).

Katere trditve so pravilne (več možnih odgovorov)?

Izberite enega ali več odgovorov:

- ☒ Možnost B je bolj primerna, če pogosto izvajamo konkatencijo (stik) dveh seznamov ✓
- ☐ Možnost B je bolj primerna, če pogosto dostopamo do elementov na naključnih pozicijah v seznamu
- ☐ Možnost B je bolj primerna, če pogosto brišemo zadnji element seznama
- ☒ Možnost B je bolj primerna, če pogosto brišemo elemente na podlagi nekega kriterija (na primer, briši vse elemente, ki so manjši od 5) ✓
- ☒ Možnost B je bolj primerna, če pogosto vstavljamo elemente na začetek seznama ✓

Vaš odgovor je pravilen.

Pravilni odgovori so: Možnost B je bolj primerna, če pogosto vstavljamo elemente na začetek seznama, Možnost B je bolj primerna, če pogosto izvajamo konkatencijo (stik) dveh seznamov, Možnost B je bolj primerna, če pogosto brišemo elemente na podlagi nekega kriterija (na primer, briši vse elemente, ki so manjši od 5)

Vprašanje **7**

Pravilno

Ocena 1,00 od 1,00

ADS LIST je, med drugim, možno implementirati kot enosmerni ali dvosmerni seznam s kazalci. Katere trditve so pravilne, če imamo pri obeh implementacijah neposreden dostop do prvega in zadnjega elementa seznama (kazalca first in last)?

Izberite enega ali več odgovorov:

- ☐ Časovna kompleksnost brisanja prvega elementa je nižja pri dvosmernem seznamu
- ☐ Časovna kompleksnost dodajanja prvega elementa je nižja pri dvosmernem seznamu
- ☐ Časovna kompleksnost dodajanja zadnjega elementa je nižja pri dvosmernem seznamu
- ☒ Dvosmerni seznam zahteva več pomnilniškega prostora kot ustrezni enosmerni seznam ✓
- ☒ Časovna kompleksnost brisanja zadnjega elementa je nižja pri dvosmernem seznamu ✓

Vaš odgovor je pravilen.

Pravilni odgovori so: Dvosmerni seznam zahteva več pomnilniškega prostora kot ustrezni enosmerni seznam, Časovna kompleksnost brisanja zadnjega elementa je nižja pri dvosmernem seznamu

◀ ADT STACK - rešitve

Skok na...

⌵

Prosojnice s kodo: rdeče-črna drevesa ▶