

# Algoritmi in podatkovne strukture – 2

## Prvi kolokvij (2015/16)

Kolokvij morate pisati posamič. Pri reševanju je literatura dovoljena. Pri odgovarjanju bodi natančni in: (i) odgovarjajte *na zastavljena* vprašanja; in (ii) odgovorite na *vsa* zastavljena vprašanja – če boste odgovarjali na vsa vprašanja, lahko dobite dodatne točke.

Čas pisanja kolokvija je 60 minut.

Veliko uspeha!

NALOGA	TOČK	OD TOČK	NALOGA	TOČK	OD TOČK
1			3		
2			4		

IME IN PRIIMEK:

---

ŠTUDENTSKA ŠTEVILKA:

---

DATUM:

---

PODPIS:

---

**1. naloga:** Butale imajo dokaj razvito gospodarstvo s celo vrsto podjetij. Po končanem šolanju se Butalci zaposlijo v različnih podjetjih in so precej ponosni na to, v katerem podjetju delajo ter se radi primerjajo med seboj, če delajo v istem podjetju. Podjetja pa se občasno prevzemajo, da optimirajo stroške in povečajo svojo konkurenčnost.

VPRAŠANJA:

- A) Katera od podatkovnih struktur, ki smo jih obravnavali pri pouku, bi bila uporabna osnova za rešitev naloge? Utemeljite odgovor.
- B) Podrobneje opišite podatkovno strukturo, ki bi pomagala Butalcem, da bi lahko učinkovito odgovarjali na poizvedbe, v katerem podjetju so zaposleni.

NAMIG: V resnici morate opisati, kako učinkovito implementirati operacije:  $SeZaposli(X, Y)$ , ki pomeni, da se je oseba  $Y$  zaposlila v podjetju  $X$ ;  $Prevzame(X, Y)$ , ki pomeni, da je podjetje  $X$  prevzelo podjetje  $Y$  in je podjetje  $Y$  nehalo obstajati; in  $ZaposlenV(X)$ , ki vrne ime podjetja, v katerem je zaposlena oseba  $X$ .

- C) Včasih želijo Butalci tudi izvedeti, kdo vse dela v podjetju  $X$ . Opišite učinkovito izvedbo takšne poizvedbe in utemeljite, kako učinkovita (časovno in prostorsko) je ter zakaj.

## 2. naloga:

VPRAŠANJA:

- A) Narišite najvišje in najnižje AVL drevo s po 15 vozlišči.
- B) Kakovost rekurzivne podatkovne strukture je definirana z najdaljšo potjo od začetka do konca strukture – na primer, od korena drevesa do njegovega najbolj oddaljenega lista ali od začetka seznama do njegovega zadnjega elementa. Opišite primer operacij nad preskočnim seznamom, ko je ta pot še posebej dolga. Utemeljite odgovor.
- C) Recimo, da imamo besedilo  $t = a_1a_2a_3\dots a_n$ , kjer so  $a_i$  iz končne abecede  $\Sigma$ . Peter Zmeda bi rad naračunal za besedilo  $t$  Parikhove vrednosti, kar pomeni, da bi rad za vsako črko  $a$  iz  $\Sigma$  preštel število njenih pojavitev v besedilu  $t$ . Recimo, v besedilu ABRAKADABRA se A pojavi petkrat in tako naprej. Očitno gre za slovar vseh črk, kjer je črka ključ in število njenih pojavitev podatek. Predlagajte učinkovito izvedbo slovarja in začrtajte algoritem, ki izračuna zahtevane vrednosti.

**3. naloga:** Min-kopica, ki smo jo srečali na predavanjih, je podatkovna struktura, pri kateri velja: (i) da je najmanjši element v korenu; (ii) da so vsi listi na isti globini ali največ na dveh; (iii) listi so levo poravnani. Min-kopica je primer implementacije vrste s prednostjo. Včasih želimo hitro dostopati ne samo do najmanjšega, ampak tudi do največjega elementa. Zato definiramo minmax-kopico<sup>1</sup>. Pri minmax-kopici veljajo naslednja pravila:

- (a) plasti v minmax-kopici štejemo od vrha navzdol in jih razdelimo na lihe (koren, njegovi vnuki in tako naprej) ter na sode (otroci korena in pravnuki korena ter tako naprej);
- (b) v lihih plasteh velja, da je v vozlišču najmanjši element podkopice ter v sodih plasteh, da je v korenu največji element kopice;
- (c) vsi listi so na isti globini ali največ na dveh; in
- (d) listi so levo poravnani.

Recimo, da imamo v minmax-kopici naslednje elemente: 1, 3, 4, 11, 12, 13, 16, 17, 24, 25, 25, 27, 30 in 77. Potem je v korenu element 1, ker je najmanjši, v enem od njegovih otrok je element 77, ker je največji, ter v drugem, recimo, 24.

VPRAŠANJA:

- A) Razvrstite zgornje elemente v minmax-kopico.
- B) Zapišite funkciji  $\text{Min}(pq)$  in  $\text{Max}(pq)$ , ki vrmeta največji in najmanjši element v kopici  $pq$ . Utemeljite pravilnost njunega delovanja in njuno časovno zahtevnost.
- C) Zapišite postopek  $\text{DelMax}(pq)$ , ki iz kopice  $pq$  izbriše največji element. Utemeljite njeno pravilnost in časovno zahtevnost.

NAMIG: Morda je preprostejša funkcija  $\text{DelMin}(pq)$ ? Če se vam zdi, zapišite ta postopek ter utemeljite njegovo pravilnost in časovno zahtevnost. Dobili boste sicer nekaj manj točk.

**4. naloga:** *Slovarji tako in drugače* Ena od oblik slovarja, ki smo jo srečali, je tudi Bloomov filter, ki pa ima samo operaciji  $\text{Insert}(\text{elt})$  in  $\text{Find}(\text{elt})$ . Bloomov filter uporablja bitni vektor, kamor shranja vse podatke. Za razliko od običajnega slovarja, lahko tukaj večkrat vstavimo isti element.

VPRAŠANJA:

- A) Tudi naš prijatelj Peter je izvedel za Bloomov filter in se je lotil njegove implementacije. Zataknilo se mu je že pri izbiri razpršilne funkcije. Po

---

<sup>1</sup>Minmax kopico smo omenjali tudi na vajah.

tehtnem premisleku in prebiranju dokumentacije je ugotovil, da je najbolje če naredi razprševanje na enak način kot pri *Cuckoo* razprševanju, ki velja za enega najboljših. Komentirajte njegovo odločitev.

- B) Če pri Bloomovem filtru nadomestimo bitni vektor z vektorjem števcov, dobimo števeni Bloomov filter. Psevdokoda vstavljanja potem izgleda takole:

```
Insert(elt):  
    for i= 1 ... k:  
        bf[ h(i, elt) ]++
```

Kjer je `bf[]` vektor števcov in imamo `k` razpršilnih funkcij. Za izračun  $i$ -te funkcije nad `elt` uporabimo klic funkcije `h(i, elt)`. Sedaj imamo nad Bloomovim filtrom namesto operacije `Find(elt)`, ki poišče element `elt`, funkcijo `Count(elt)`, ki vrne kolikokrat je bil element `elt` vstavljen v Bloomov filter. Zapišite psevdokodo funkcije `Count(elt)`, utemeljite njeno pravilnost ter časovno in prostorsko zahtevnost.

- C) Pri binomski vrsti s prednostjo smo omenjali, da za učinkovito izvedbo operacij `DecreaseKey(&elt, d)` in `Delete(&elt)` potrebujemo neposredno referenco na element v strukturi (zgoraj označeno z `&`). Kaj pa, če referenc nimamo in moramo implementirati omenjeni funkciji tako, da dobimo kot parameter sam element `elt`? Predlagajte in utemeljite rešitev.

NAMIG: Za reševanje tega vprašanja *ne* potrebujemo Bloomovega filtra.