

Algoritmi in podatkovne strukture – 2

Prvi kolokvij (2012/13)

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 izpita 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: *Uvod in osnove.* Imamo naslednji program:

```
int foo (int x, y)
    if x == 0 return y
    y= 2*y + x % 2
    return foo (x \ 2, y)
```

VPRAŠANJA:

1. Sledite izvajanju funkcije za naslednje klice: $\text{foo}(5, 0)$, $\text{foo}(6, 0)$ in $\text{foo}(7, 0)$.
2. Kakšna je časovna zahtevnost programa pri klicu $\text{foo}(n, 0)$? Utemeljite svoj odgovor.
3. Imamo funkciji $f_1(n) = 12n + 11 \lg n$ in $f_2(n) = 11/n + 12 \lg n$. Katerim izmed naslednjih družin pripada vsaka od funkcij: $O(n)$, $\Omega(n)$, $\Theta(n)$, $O(\log n)$, $\Omega(\log n)$ in $\Theta(\log n)$. Utemeljite svoj odgovor.

2. naloga: Peter Zmeda in njegova prijateljica Špela se igrata naslednjo igrico:

1. na začetku imamo n lučk, ki so označene od 1 do n ;
2. Peter k krat poveže po dve lučki;
3. Špela se z žico dotakne ene od lučk in potem vse lučke, ki jih je Peter povezal z dotaknjeno lučko, zasvetijo.

VPRAŠANJA:

1. Naj bo $n = 10$ in Peter nato poveže pare: $(5, 7)$, $(1, 8)$, $(2, 9)$, $(3, 8)$, $(4, 7)$ in $(7, 1)$. Špela se z žico dotakne lučke 4. Katere od naslednjih lučk svetijo: 2, 5, 8, 9 in 10?
2. Pri velikih n postane igrice precej težko izvedljiva, zaradi česar sta se Špela in Peter odločila napisati računalniški program, ki bo simuliral igrico. Program in posledično podatkovna struktura mora nuditi naslednje ukaze:
 - `Povezi(x, y)`, ki poveže lučki x in y ;
 - `Prizgi(x)`, s katerim se dotaknemo lučke x in
 - `Sveti(x)`, ki vrne DA ali NE odvisno od tega, ali lučka x sveti.

Opišite kako naj izgleda podatkovna struktura, ki podpira opisano igrico ter kako so implementirani posamezni ukazi.

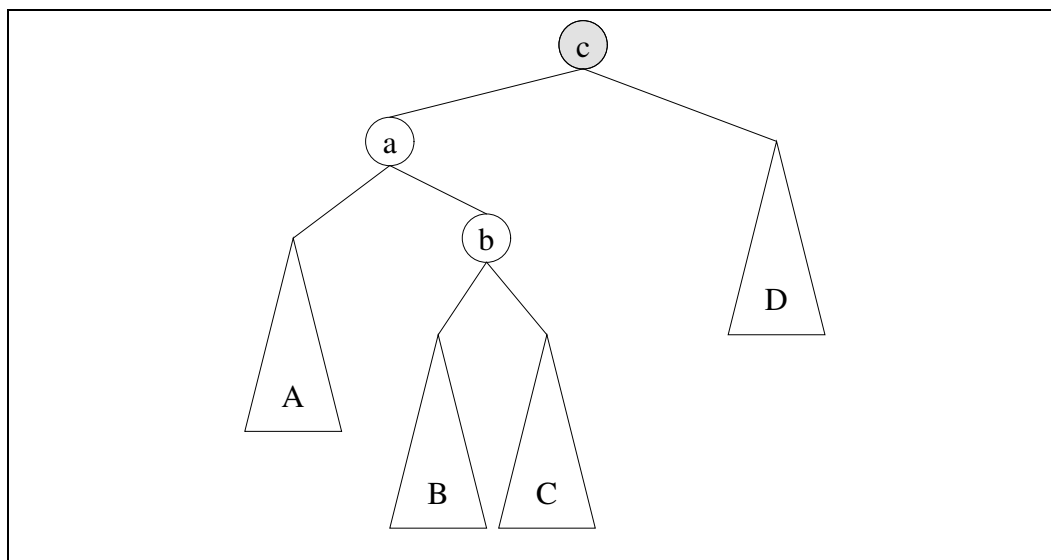
NAMIG: Morda lahko uporabite kakšno znano podatkovno strukturo.

3. Opišite, kako bi shranili popolno k -tiško drevo kot implicitno podatkovno strukturo. Z drugimi besedami, kako bi shranili omenjeno drevo v polju. Pri tem morate opisati, kako od vozlišča shranjenega na indeksu $v[i]$:

- pridemo do vozlišča, ki predstavlja njegovega j -tega naslednika ($1 \leq j \leq k$); in
- kako pridemo do njegovega starša.

NAMIG: Polje naj se prične na indeksu 1. Koren drevesa je $v[1]$, prvi naslednik na $v[2]$, drugi na $v[3]$ in tako naprej. Pri kopici (*heap*) je uporabljena podobna preslikava, le da je tam $k = 2$.

3. naloga: V tem vprašanju bomo obravnavali rdeče-črno drevo. Na sl. 1 je kon-



Slika 1: Rdeče-črno drevo po vstavljanju v poddrevesu B.

figuracija RČ drevesa po tem, ko smo v poddrevo B vstavili element ter sta vozlišči a in b rdeči ter vozlišče c črno. Poleg tega so koreni poddreves A, B, C in D črni ter tudi črne višine poddreves enake.¹

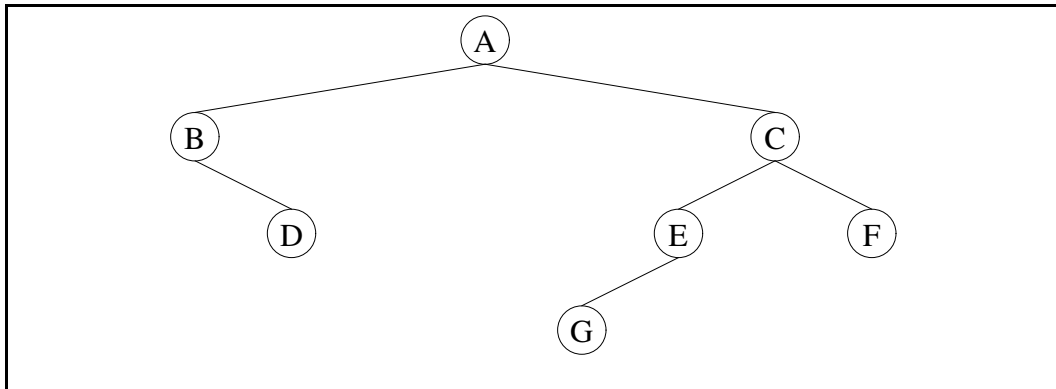
VPRAŠANJA:

1. Kako, če je potrebno, se naj spremeni oblika drevesa na sl. 1. Narišite in utemeljite odgovor.
2. Zapišite psevdokodo, ki opravi omenjeno preoblikovanje.

¹Črna višina poddrevesa je število črnih vozlišč od korena do poljubnega lista.

3. B drevo. Naš prijatelj Peter Zmeda je pregledoval kodo za brisanje iz B drevesa in zazdelo se mu je, da bi lahko bila učinkovitejša, če bi dovolil, da je v vsakem vozlišču namesto vsaj $\frac{b}{2}$ elementov $\frac{b}{3}$ elementov. Komentirajte njegovo idejo. Pri tem se osredotočite na zapletenost kode, velikost zasedenega pomnilnika in učinkovitost operacij.

4. naloga: Imamo drevo, kot ga prikazuje sl. 2.



Slika 2: Primer dvojiškega drevesa.

VPRAŠANJA:

1. Izpišite zaporedje vozlišč, ki jih dobimo pri vmesnem obhodu (*inorder*) drevesa na sl. 2.
2. Drevo lahko rekonstruiramo, če imamo na voljo dva obhoda. Naj bo zaporedje vozlišč, ki jih dobimo pri vmesnem obhodu $v[1 \dots n]$ in zaporedje vozlišč, ki jih dobimo pri prvem (*preorder*) obhodu $p[1 \dots n]$. Zapišite algoritem, ki iz $v[\dots]$ in $p[\dots]$ rekonstruira drevo. Pseudokoda bo povsem dovolj. Na primer:
 - Vozlišče $p[1]$ je koren drevesa in vozlišča \dots so v levem ter \dots v desnem poddrevesu.
 - Potem je koren levega \dots
3. Utemeljite, zakaj zapis samo enega obhoda ni dovolj za rekonstrukcijo drevesa.