

Algoritmi in podatkovne strukture – 2

Prvi kolokvij (2010/11)

Kolokvij morate pisati posamič. Pri reševanju je literatura dovoljena.
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 naslednjo funkcijo:

```
int bla (int n) {  
    if n == 0 return 0  
    if n == 1 return 1  
    return bla(n-1) + bla(n-2) + 1  
}
```

VPRAŠANJA:

1. Kaj funkcija izpiše za naslednje n : 5, 6, 7, 8, 9, 10 in 11.
2. Zapišite rekurzivno enačbo za časovno zahtevnost funkcije `bla` v odvisnosti od n in jo utemljite.
3. Izračunajte časovno in prostorsko zahtevnost funkcije v obliki $O(\dots)$.

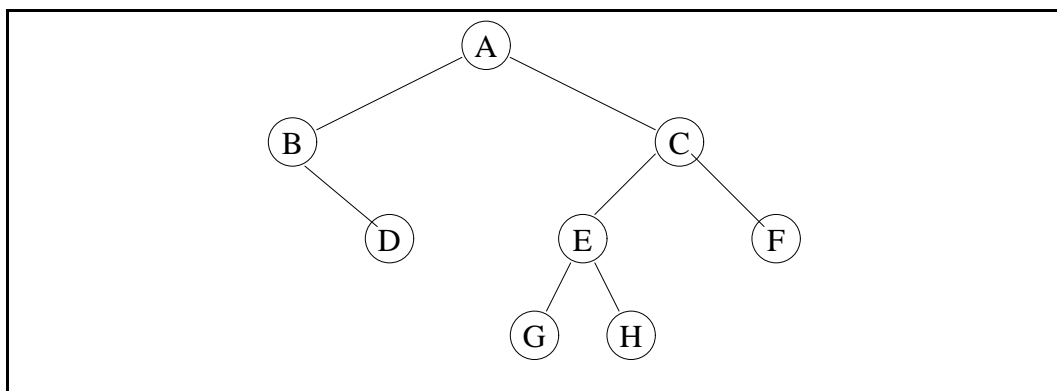
NAMIG: Pri dokazu časovne zahtevnosti najprej ocenite zahtevnost in nato uporabite indukcijo. Časovna (!) zahtevnost je večja kot polinomska.

4. [DODATNA] Na predavanjih smo večkrat omenjali, da lahko pospešimo kakšen algoritem, če uporabimo več prostora. Ali lahko to naredite za zgornji primer?

2. naloga: *Drevesa in disjunktne množice.* Na predavanjih smo omenjali premi, vmesni in obratni obhod drevesa.

VPRAŠANJA:

1. Na sliki sl. 1 imamo neko binarno drevo. Izpišite vozlišča ob obratnem



Slika 1: Primer dvojiškega (ne-iskalnega) drevesa.

(*postorder*) obhodu. Pokažite, da takšen izpis, ki ga tvorimo ob obratnem obhodu, ne določa enolično topologije (oblike) drevesa.

NAMIG: Poiščite še kakšno drevo, ki tvori enako zaporedje vozlišč ob obratnem obhodu.

2. Kateri od omenjenih obhodov ima smisel pri številskih drevesih (*tries*)? Utemeljite odgovor.
3. V prvem vprašanju smo omenili, da samo iz obratnega obhoda ne moremo rekonstruirati oblike drevesa. Da pa se pokazati, da iz izpisa premega in obratnega obhoda pa se da. Pokažite to trditev oziroma predstavite algoritem.

NAMIG: Poskusite najprej z drevesom s sl. 1.

3. naloga: Slovar in zagotovljeni čas.

VPRAŠANJA:

1. Tokrat bomo naredili premi obhod dvojiškega iskalnega drevesa. Kakšne vrednosti dobimo kot rezultat obhoda?
2. Peter Zmeda je slišal, da iz urejenega polja dolžine n lahko zelo preprosto v linearnem času naredi uravnoteženo dvojiško iskalno drevo, žal pa je pozabil podrobnosti. Napišite psevdokodo algoritma, ki kot vhodni podatek dobi polje $a[0 \dots n-1]$, ki v linearnem času ustvari uravnoteženo dvojiško iskalno drevo.

NAMIG: Lotite se dela rekurzivno: najprej koren, nato levo poddrevo ter na koncu še desno poddrevo.

3. [DODATNA] Ali se vam iz odgovorov na zgornji vprašanje porodi kakšna zamisel?

4. naloga: Slovar in pričakovani čas. Recimo, da imamo naslednje ključe: 24, 1, 15, 10, 14, 23, 18, 19, 6, 5 in 21. Ključne boste vstavili v razpršilno tabelo in prešteli število vseh dostopov v tabelo, ki jih boste morali opraviti pri tem. To številko potem delite s številom vstavljenih elementov in boste dobili povprečno število dostopov.

VPRAŠANJA:

1. Najprej vstavite ključe v razpršilno tabelo velikosti $m = 15$, pri čemer je tabela na začetku prazna. Kot razpršilno funkcijo uporabite funkcijo $h(k) = k \bmod m$. V primeru sovpadanja uporabiti odprto naslavljanje z linerano funkcijo. Prikažite (i) posamezne korake pri vstavljanju elementov, (ii) podajte skupno in povprečno število dostopov do podatkovne strukture, (iii) največje in najmanjše število dostopov pri vstavljanju enega elementa ter (iv) velikost celotne podatkovne strukture.
2. Nato vstavite ključe v razpršilno tabelo velikosti $m = 7$, pri čemer je tabela na začetku prav tako prazna. Kot razpršilno funkcijo uporabite ponovno funkcijo $h(k) = k \bmod m$ (m je sedaj drugačen od prejšnjega vprašanja). V primeru sovpadanja uporabiti veriženje z neurejenim seznamom. Ponovno prikažite (i) posamezne korake pri vstavljanju elementov, (ii) podajte skupno in povprečno število dostopov do podatkovne strukture, (iii) največje in najmanjše število dostopov pri vstavljanju enega elementa ter (iv) velikost celotne podatkovne strukture.
3. Primerjajte in komentirajte rezultate obeh vprašanj.