

Algoritmi in podatkovne strukture – 2

Pisni izpit 26. rožnik 2017 (2016/17)

Pisni izpit 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.

Čas pisanja izpita je 90 minut.

Veliko uspeha!

| NALOGA | TOČK | OD TOČK | NALOGA | TOČK | OD TOČK |
|--------|------|---------|--------|------|---------|
| 1 | | | 3 | | |
| 2 | | | 4 | | |

IME IN PRIIMEK: _____

ŠTUDENTSKA ŠTEVILKA: _____

DATUM: _____

PODPIS: _____

1. naloga: Imamo polje celih števil $A[1..n]$. Implementirati želimo funkcijo $Sodih(i, j)$, ki vrne število sodih števil v polju med indeksoma i in j vključno. Za

[97, 38, 82, 86, 10, 51, 38, 88, 79, 55, 45, 40] (1)

vrne $Sodih(1, 3)$ vrednost 2, saj sta števili 38 in 82 sodi, medtem ko vrednost 7 vrne $Sodih(-100, +100)$.

VPRAŠANJA:

- A) (i) Zapišite algoritem $Sodih()$, ki deluje nad globalno spremenljivko A in pri tem ne uporablja predprocesiranja. (ii) Kakšna je njegova časovna zahtevnost.
- B) Tokrat je dovoljeno predprocesiranje, kar dovoljuje izgradnjo dodatne podatkovne strukture. (i) Opišite dodatno podatkovno strukturo, ki bo omogočala poizvedbe v času $O(1)$. (ii) Kakšna je časovna zahtevnost predprocesiranja in utemeljite časovno zahtevnost $O(1)$ za poizvedbe. (iii) Kakšna je prostorska zahtevnost vaše dodatne podatkovne strukture.
- C) Sedaj problem razširimo tako, da poleg funkcije $Sodih()$ dodamo še funkciji $Vstavi(i, x)$ in $Izloci(i)$. Prva v polje A na mesto i vstavi element x . Na primer iz polja iz (1) po klicu $Vstavi(3, 26)$ dobimo polje

[97, 38, 82, 26, 86, 10, 51, 38, 88, 79, 55, 45, 40]

in če sedaj pokličemo še funkcijo $Izloci(6)$, dobimo

[97, 38, 82, 26, 86, 51, 38, 88, 79, 55, 45, 40] .

(i) Opišite podatkovno strukturo in vse tri funkcije nad njo. (ii) Kakšna je prostorska zahtevnost vaše podatkovne strukture? (iii) Kakšna je časovna zahtevnost vaše rešitve (vse tri operacije)?

NAMIG: Učinkovitejša kot bo vaša rešitev, več točk boste prejeli.

2. naloga: Statistika. Ena ključnih operacij za izdelavo statistike je štetje velikosti vzorca. V tej nalogi boste predlagali podatkovne strukture, ki nam bodo omogočale dovolj dobro štetje velikosti vzorca in delovati bodo morale učinkovito.

VPRAŠANJA:

- A) V Sloveniji imamo 11 registrskih območij (MS, MB, CE, NM, KK, SG, LJ, KR, PO, GO in KP). Na cesti bi radi šteli koliko vozil iz posameznega registrskega območja se je peljalo mimo našega števca. (i) Predlagajte ustrezno podatkovno strukturo za štetje. (ii) Utemeljite odgovor. (iii) Kakšna

je prostorska zahtevnost vaše strukture? (iv) Kakšna je časovna zahtevnost operacij `Pristej(oznaka)`, ki poveča števec za oznako `oznaka` za 1, in `Koliko(oznaka)`, ki vrne trenutno vrednost števca za oznako `oznaka`.

- B) Tokrat bomo šteli opazovali promet na požarni pregradi. Točneje, radi bi vedeli, če je do požarne pregrade prišel kakšen paket z določenega IP naslova. (i) Ponovno predlagajte ustrezno podatkovno strukturo za štetje. (ii) Utemeljite odgovor. (iii) Kakšna je prostorska zahtevnost vaše strukture? (iv) Kakšna je časovna zahtevnost operacij `Prišel(ip)`, ki v strukturi zabeleži, da je prišel paket z danega `ip` naslova, in `JePrišel(oznaka)`, ki vrne ali je prišel paket z danega `ip` naslova.
- C) (i) Nadgradite podatkovno strukturo iz prejšnjega vprašanja tako, da bo tudi štela število paketov, ki pridejo z določenega `ip` naslova. (ii) Kakšna je prostorska zahtevnost vaše strukture? (iii) Kakšna je časovna zahtevnost operacij `Pristej(ip)` in `Stevilo(ip)`.

3. naloga: V Butalah so čudni tiči in, ker so čudni, imajo tudi čudne vrednosti kovancev. Tako imajo kovanec za 31, za 17, za 10 in za 1 BUT. Na solomatu (da, to je avtomat, kjer kupuješ butalsko sol in njene izdelke) je običajno potrebno vrniti drobiž, če stranka plača več, kot je cena izdelka. Nalogo, kakšne kovanec naj solomat vrne, so zaupali Petru Zmedi in mu pri tem zabičali, da naj njegov algoritem vrne najmanjše število kovancev. Peter je zasnoval naslednji algoritem:

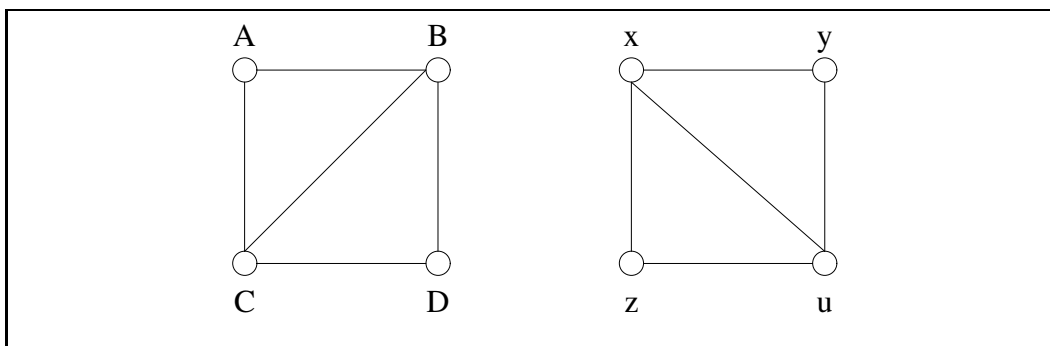
```
Vrni(znesek):
    while znesek > 31:
        vrni kovanec za 31 BUT
        znesek = znesek - 31
    while znesek > 17:
        vrni kovanec za 17 BUT
        znesek = znesek - 17
    while znesek > 10:
        vrni kovanec za 10 BUT
        znesek = znesek - 10
    while znesek > 1:
        vrni kovanec za 1 BUT
        znesek = znesek - 1
```

VPRAŠANJA:

- A) (i) Kakšno tehniko je uporabil Peter pri načrtovanju svojega algoritma. Utemeljite odgovor. (ii) Izpišite, koliko kovancev in katerih vrne Petrov algoritem pri zneskih 30 BUT, 50 BUT in 70 BUT.

- B) Źal Petrov algoritem ne vrne najmanjšega števila kovancev pri vsakem znesku.
 (i) Poiščite najmanjši znesek, pri katerem Petrov algoritem ne deluje pravilno.
 (ii) Kako bi opisali zneske, pri katerih Petrov algoritem odpove?
- C) (i) Sestavite algoritem, ki vedno naračuna najmanjše število kovancev, ki jih moramo vrniti. (ii) Utemeljite pravilnost vašega algoritma. (iii) Kakšna je časovna in kakšna prostorska zahtevnost vašega algoritma.

4. naloga: Grafi. Grafi lahko izgledajo različno, a v resnici niso, kot je preprost primer na sl. 1.



Slika 1: Dva enaka grafa, ki sta narisana različno.

VPRAŠANJA:

- A) (i) Zapišite definicijo, kdaj sta grafa enaka. (ii) Utemeljite odgovor.
- B) Peter Zmeda ima dva grafa $G_1(V_1, E_1)$ in $G_2(V_2, E_2)$, za katera trdi, da sta enaka. (i) Kakšno dokazilo naj vam poda, da mu boste verjeli? Utemeljite odgovor. (ii) Koliko časa boste potrebovali (časovna zahtevnost), da se prepričate v pravilnost njegove trditve?
- C) Pavel si je zamislil naslednji algoritem, kako bi ugotovil ali sta grafa enaka:

Enaka (G_1, G_2):

Uredi V_1 glede na število sosednjih vozlišč,
 če je več vozlišč z enakim številom sosedov,
 uredi po oznakah vozlišč

Uredi enako V_2

Preimenuj vozlišča V_1 v oznake $1, \dots, n$

Preimenuj vozlišča V_2 v oznake $1, \dots, n$

Preveri ali sta grafa enaka

Ali je njegov algoritem pravilen? Utemeljite odgovor.