

Algoritmi in podatkovne strukture – 2

Pisni izpit 27. rožnik 2012 (2011/12)

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: Tokrat bomo implementiral podatkovno strukturo sklad. Recimo, da imamo že naslednji del implementacije v jeziku podobnem Javi:

```
public class Stack {  
    private int[] data;  
    void Stack(int size) {  
        data= new int[size];  
        ...  
    }  
    Boolean Empty() { ... }  
    Boolean Full() { ... }  
    int Top() { ... }  
    void Push() { ... }  
    int Pop() { ... }  
}
```

VPRAŠANJA:

1. Implementirajte vse metode, ki so označene s pikicami. Pri tem bo potrebno morda dodati še kakšne podatke v razred.
2. Kakšna je prostorska zahtevnost podatkovne strukturev odvisnosti od števila elementov v skladu? Utemeljite odgovor.

NAMIG: Bodite pozorni, da govorimo o številu elementov, ki so shranjeni na skladu.

3. peter Zmeda bi rad nadgradil zgornjo podatkovno strukturo tako, da metoda Full ne bi nikoli vrnila true – z drugimi besedami, da bi se sklad nikoli ne napolnil. Pomagajte mu. Najprej opišite idejo z besedami in šele nato se lotite kodiranja.

DODATNA Kakšna je prostorska zahtevnost popravljene podatkovne strukture v odvisnosti od števila elementov v skladu? Utemeljite odgovor.

NAMIG: Točke boste dobili glede na učinkovitost vaše rešitve. Če boste opisali smiselno rešitev, boste dobili že polovico točk.

2. naloga: Vrste s prednostjo.

VPRAŠANJA:

1. Recimo, da imamo v vrsti s prednostjo 2012 elementov. Odgovorite na naslednja vprašanja za različne implementacije vrste s prednostjo in vsakega od odgovorov utemeljite: i) Kako visoka je dvojiška kopica? ii) Kako visoko je najvišje drevo binomske kopice? iii) Kako visoko je največ najvišje drevo Fibonaccijeve kopice? iv) Ali je možno, da je najvišje drevo Fibonaccijeve kopice visoko 1?
2. Pri Dijkstrovem algoritmu uporabljamo vrsto s prednostjo. Katero od zgoraj navedenih implementacij priporočate in zakaj?

NAMIG: Recimo, da nas zanima predvsem časovna zahtevnost.

3. Nad Fibonaccijevo kopico po vrsti izvedite naslednje operacije in sproti izrisujte podatkovno strukturo (I pomeni vstavi, M minimum in DM zbrši najmanjši element):

I 17, I 3, I 5, I 1, M, I 10, DM, I 4, DM, DM

3. naloga: Pri računalništvu poznamo pojem konjunktivne oblike izraza, ki pomeni konjunkcijo disjunkcij. Posebna oblika konjunktivne oblike je takšna oblika, ki ima v vsaki disjunkciji natančno dve spremenljivki.

VPRAŠANJA:

1. Problem iskanje takšnih vrednosti posameznih spremenljivk (`true`, `false`), da izračunajo celoten izraz v `true`, pravimo zadovoljitev (*satisfiability*, SAT). Posebno obliko problema SAT, kjer sta v disjunkciji samo dve spremenljivki, imenujemo 2-SAT. V izrazu

$$(x_1 \vee \overline{x_2}) \wedge (x_2 \vee \overline{x_3}) \wedge (x_3 \vee \overline{x_1})$$

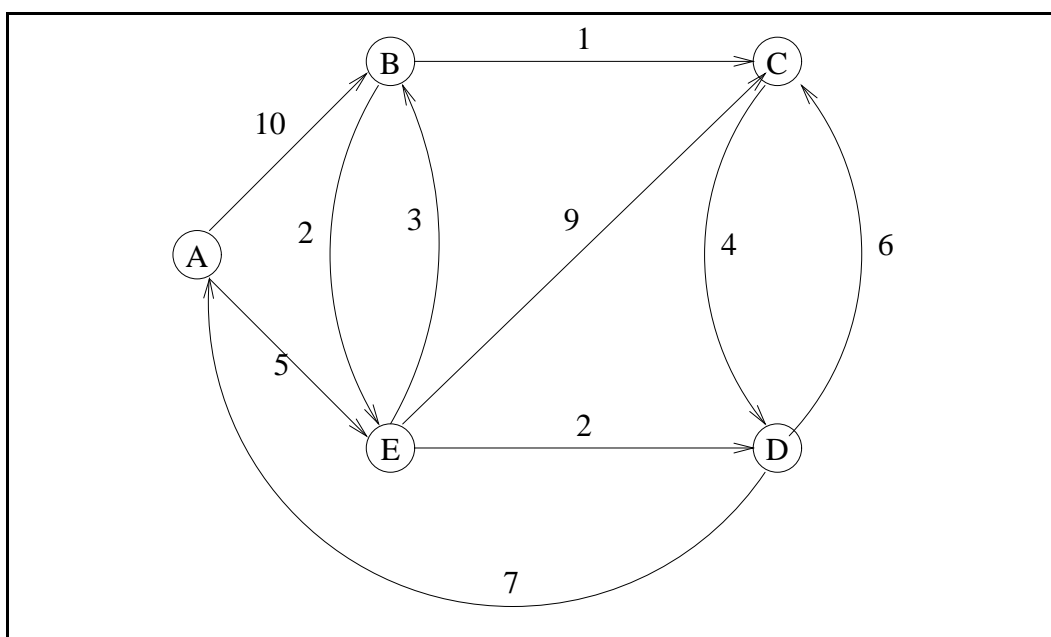
določite poiščite vse vrednosti spremenljivk x_1 , x_2 in x_3 , ki izračunajo izraz na `true`.

2. Poiščite algoritem, ki bo v splošnem poiskal vrednosti spremenljivk x_i tako, da se bo splošni 2-SAT izraz naračunal na `true`.

NAMIG: Razmislite, kam vas pripelje požrešni algoritem?

3. Kakšna je časovna zahtevnost vašega algoritma glede na število spremenljivk?

NAMIG: Za zvedave: 2-SAT se dá prevesti na problem iskanja cikla v grafu. Po izpitu poiščite rešitev na spletu in se oglasite na forum!



Slika 1: Primer usmerjenega grafa.

4. naloga: Pri tej nalogi bomo imeli opravka z grafi. Na zadnjem izpitu smo imeli usmerjen graf s sl. 1 Pravimo, da graf vsebuje *cikel*, če obstaja usmerjena pot, ki nas iz nekega vozlišča u pripelje nazaj v vozlišče u . Ciklu, na katerega poti se nobeno vozlišče ne ponovi, pravimo *preprosti cikel*.

VPRAŠANJA:

1. Poiščite v grafu s sl. 1 najdaljši preprosti cikel. Utemeljite, da je najdeni preprosti cikel res najdaljši.
2. Opišite algoritem, ki v danem grafu $G(V, E)$ preveri, ali obstaja cikel.
3. Če pričnemo iskati cikel v nekem vozlišču u , ali mora u biti del cikla? Utemeljite odgovor.

NAMIG: Graf s sl. 1 ni dober primer za iskanje odgovora na to vprašanje.

4. Kakšna je časovna zahtevnost vašega algoritma?