

Algoritmi in podatkovne strukture 1, PISNI IZPIT, 10.2.2014

Od literature je dovoljen samo lastnoročno z navadnim svinčnikom napisan list A4 in podpisan s kemičnim svinčnikom, ki ga je potrebno oddati skupaj z izpitom. Naloge so enakovredne. Čas pisanja 90 min. Komentirajte programe! Ustni izpit za tiste, ki želijo popravljati oceno: v torek, 17.2.ob 10h v kabinetu prof. Kononenka (R2.07).

1. Dan je algoritem:

```
class Mnozica { Data en; Mnozica nasl ; }

boolean ele(Data x, Mnozica a) {
    if (a == null) return false ;
    else if (a.en.compareTo(x) == 0) return true ;
    else return ele(x,a.nasl)
} // ele
```

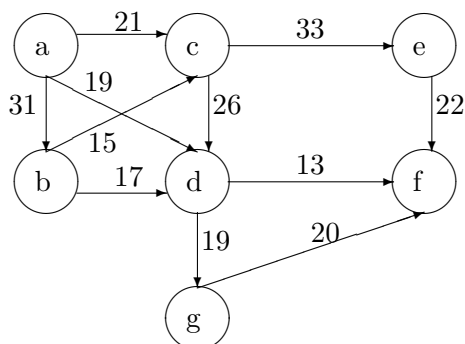
- (a) Definiraj podatkovno strukturo seznama množic in razvij algoritem, ki iz dane množice S in danih elementov $x, y \in S$ tvori potenčno množico, iz katere so izvzete vse podmnožice originalne množice S , ki vsebujejo dani element x in ki ne vsebujejo danega elementa y .
- (b) Izberi ustrezne parametre in oceni časovno zahtevnost algoritma.
2. Za dani program so bili izmerjeni naslednji časi izvajanja za različne velikosti vhodnih podatkov:

velikost	5	10	15	20
čas	25	2650	9775	23650

Katera funkcija najbolj ustreza zahtevnosti tega programa v odvisnosti od velikosti vhodnih podatkov (odgovor argumentiraj z oceno konstant v enačbi $T(n) = a f(n) + c$): $O(\log n)$, $O(n)$, $O(n \log n)$ ali $O(n^3)$?

3. Dan je graf na sliki. Na grafu simuliraj algoritma:

- (a) Za iskanje kritične poti od a do f po principu dinamičnega programiranja, tako da zapišeš zaporedje seznama vozlišč z nepregledanimi nasledniki.
- (b) Dijkstra za iskanje drevesa najkrajših poti iz začetnega vozlišča a , tako da narišeš zaporedje kopic in na koncu celotno zgrajeno drevo najkrajših poti.



4. Na začetku je drevo levo poravnano in vsebuje vsa naravna števila od vključno 19 do vključno 28. Za dano zaporedje vstavljanja in brisanja elementov:
vstavi: 29, 1;
briši: 1, 19
nariši 2 drevesi: eno po končanem vstavljanju in eno po končanem brisanju, če operacije izvašaš na:
a) binarnem iskalnem drevesu ;
b) AVL-drevesu;
c) kopici (pri kopici privzami tako začetno stanje, da je desni brat vedno za ena večji od levega brata).
5. Spremeni algoritem *ele* iz 1. naloge v iterativnega, postavi ustrezen začetni in ciljni pogoj ter dokaži:
a) parcialno pravilnost iterativne verzije programa *ele* in
b) totalno pravilnost iterativne verzije programa *ele*.