

Diskretne i kombinatorne metode za računarsku grafiku

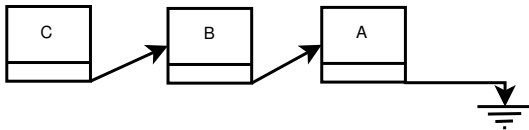
```

1: procedure NEKI SORT(A)
2:   n ← length(A)
3:   for i ← 1 to n - 1 do
4:     i_min ← i
5:     for j ← i + 1 to n do
6:       if A[j] < A[i_min] then
7:         i_min ← j
8:       end if
9:     end for
10:    if i ≠ i_min then
11:      swap(A[i], A[i_min])
12:    end if
13:    writeln(A)
14:  end for
15: end procedure

```

1. Propustiti ulaz [2, 8, 14, 8, 1, 3] kroz algoritam SELECTION SORT i ispisati stanje niza A koje se ispisuje u liniji 13.
2. Za ulazni niz [2, 8, 14, 8, 1, 3], koliko će puta upoređivanje u liniji 6 biti izvršeno, a koliko puta zamena (swap) u liniji 11?
3. Za obrnuto sortirani ulazni niz A dimenzije n, koliko će puta upoređivanje u liniji 6 biti izvršeno, a koliko puta zamena (swap) u liniji 11?
4. Dati definiciju "velikog Θ" ponašanja i pokazati da je broj upoređivanja iz zadatka 3 reda $\Theta(n^2)$.

5. Napisati program u programskom jeziku C koji pravi povezanu listu sa slike, zatim ispisuje njen sadržaj, i na kraju oslobađa dinamički alociranu memoriju.



Koristiti tip podataka cvor:

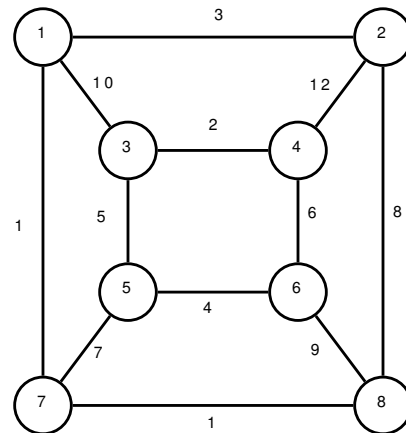
```

typedef struct _cvor cvor;
struct _cvor
{
    char podatak;
    cvor *sledeci;
};

```

6. Za graf sa slike desno napisati reprezentaciju listama susedstva. Ignorirati težine grana, držati se leksikografskog redosleda. Napisati proceduru STEPEN(i) koja koristeći reprezentaciju listom susedstva nalazi stepen za čvor i. U proceduri STEPEN pretpostaviti da je graf zadat nizom pokazivača G[i] na povezane liste susedstva (kao iz prethodnog zadatka).

Napisati redosled kojim su dodavane grane.



7. Za graf sa slike desno naći minimalno pokrivajuće drvo Kruskalovom metodom.

8. Na graf sa slike gore primeniti DFS algoritam. Dati crtež grafa sa napisanim d i f vrednostima pored čvorova i tipom grane (T/B/F/C) na granama. Ignorirati težine grana i držati se leksikografskog redosleda.