Fakultet tehničkih nauka, Novi Sad
Animacija u inženjerstvu

28. IV 2023. godine

Diskretne i kombinatorne metode za računarsku grafiku

```
1: procedure SELECTION 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. Primeniti algoritam NEKI SORT na ulaz $A = [2, 8, 14, 8, 1, 3]$ i ispisati stanje niza A koje se ispisuje u liniji 13.

   1 8 14 8 2 3

   1 2 14 8 8 3

   1 2 3 8 8 14

   1 2 3 8 8 14

   1 2 3 8 8 14

2. Za ulazni niz $[2, 8, 14, 8, 1, 3]$, koliko će puta poređenje u liniji 6 biti izvršeno, a koliko puta zamena (swap) u liniji 11?

   poređenja (linija 6): $5 + 4 + 3 + 2 + 1 = 15$ puta

   zamene - swap (linija 11): 3 puta

3. Za obrnuto sortirani ulazni niz A dužine n, koliko će puta poređenje u liniji 6 biti izvršeno, a koliko puta zamena (swap) u liniji 11?

   poređenja (linija 6): $n - 1 + n - 2 + \cdots + 1 = n(n-1)/2$ puta

   zamene - swap (linija 11): $\lfloor n/2 \rfloor$ puta jer će se parovi po jednom zameniti

4. Dati definiciju "velikog Θ" ponašanja i pokazati da je broj poređenja iz zadatka 3 reda $\Theta(n^2)$.

   $$\Theta(g) = \{ f | (\exists c_1 > 0)(\exists c_2 > 0)(\exists n_0 \in \mathbb{N})(\forall n)(n \geq n_0) \Rightarrow (0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n))\}$$

   Treba pokazati da je za neko $c_1, c_2 > 0, n_0 \in \mathbb{N}$, $\forall n$

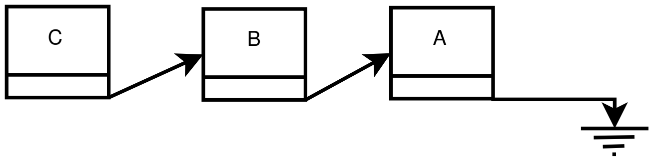
   $$n \geq n_0 \Rightarrow 0 \leq c_1 n^2 \leq \frac{n(n-1)}{2} \leq c_2 n^2.$$

   Za desnu nejednakost je dovoljno uzeti $c_2 = \frac{1}{2}$. Da bismo našli c_1, podelimo levu nejednakost sa n^2. Dobijamo

   $$0 \leq c_1 \leq \frac{1}{2} - \frac{1}{2n}, \text{ odakle } n \geq 2 =: n_0.$$

   Sad možemo uzeti c_1 koji zadovoljava $0 \leq c_1 \leq \frac{1}{2} - \frac{1}{2 \cdot 2} = \frac{1}{4}$, recimo $c_1 := \frac{1}{4}$.

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 procedure push i pop iz biblioteke stack.




```
#include "stack.h"
void main(void)
{

// Ovaj program napisati

}
```

```
#include "stack.h"

int main(void)
{
    stack S;

    makenull(&S);

    push(&S, 'A');
    push(&S, 'B');
    push(&S, 'C');

    while(!isempty(S))
        printf("%c\n",pop(&S));

    return 0;
}
```

6. Za graf sa slike desno napisati reprezentaciju listama susedstva. Ignorisati težine grana, držati se leksikografskog redosleda.

Napisati proceduru STEPEN(i) koja, koristeći reprezentaciju listom susedstva, nalazi stepen za čvor i.

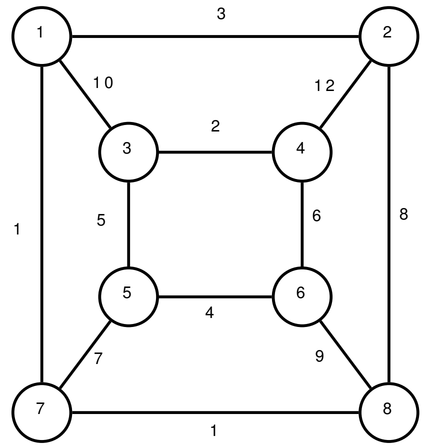
U proceduri STEPEN pretpostaviti da je graf zadat nizom pokazivača $G[i]$ na povezane liste susedstva (kao iz prethodnog zadatka).

i	$Adj(i)$		
1	2	3	7
2	1	4	8
3	1	4	5
4	2	3	6
5	3	6	7
6	4	5	8
7	1	5	8
8	2	6	7

```
int stepen(cvor *G[], int i)
{
    cvor *gr = G[i];
    int s = 0;
    while(gr){
        s++;
        gr = gr->sledeci;
    }
    return s;
}
```

7. Za graf sa slike desno naći minimalno pokrivajuće drvo Kruskalovom metodom. Napisati redosled kojim su dodavane grane.

(u,v)	w
$(1,7)$	1
$(7,8)$	1
$(3,4)$	2
$(1,2)$	3
$(5,6)$	4
$(3,5)$	5
$(5,7)$	7
$\sum$	23



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.

Ignorisati težine grana i držati se leksikografskog redosleda.