Fakultet tehničkih nauka, Novi Sad
Animacija u inženjerstvu

2. II 2016. godine

Diskretne i kombinatorne metode za računarsku grafiku

1. Napisati algoritam za sortiranje umetanjem, takozvani Insertion sort.

```
procedure Insertion sort(A)
    for j ← 2 to length(A) do
        key ← A[j]
        i ← j − 1
        while i > 0 & A[i] > key do
            A[i + 1] ← A[i]
            i ← i − 1
        end while
        A[i + 1] ← key
    end for
end procedure
```

Neka je $C(n)$ broj poređenja ključeva ulaznog niza koja se vrše pri sortiranju algoritmom Insertion sort.

2. Koliko iznosi $C(n)$ za ulazni niz $A = [3, 5, 1, 6, 4, 2]$?

$$C(6) = 1 + 2 + 1 + 3 + 5 = 12\,.$$

3. Koliko iznosi $C(n)$ za niz dužine n koji je obrnuto sortiran?

$$C(n) = 1 + 2 + \cdots n - 1 = \frac{n(n-1)}{2}\,.$$

4. Dati definiciju velikog Θ ponašanja i pokazati da je $n^2 + n + 1 = \Theta(n^2)$.

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

   Umesto da pišemo $f \in \Theta(g)$, pišemo $f = \Theta(g)$ i čitamo: funkcija f se ponaša kao $\Theta(g)$ (kao veliko theta od g).

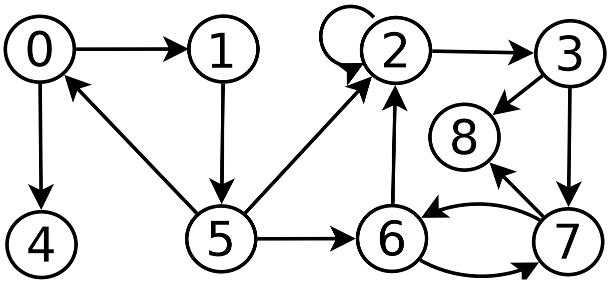
   Možemo uzeti $c_1 = 1, c_2 = 3, n_0 = 1$, jer za $n \geq n_0$ važi:

$$1 \leq 1 + \frac{1}{n} + \frac{1}{n^2} \leq 3 \;\Leftrightarrow\; c_1 \leq 1 + \frac{1}{n} + \frac{1}{n^2} \leq c_2 \;\Leftrightarrow\; c_1 n^2 \leq n^2 + n + 1 \leq c_2 n^2\,.$$

   Da li je $\sqrt{n^5} + n = \Theta(n^2)$? Ne.

   Da li je $\sqrt[5]{n^2} + n = \Theta(n^2)$? Ne.

   Da li je $C(n) = \Theta(n^2)$? Ne.



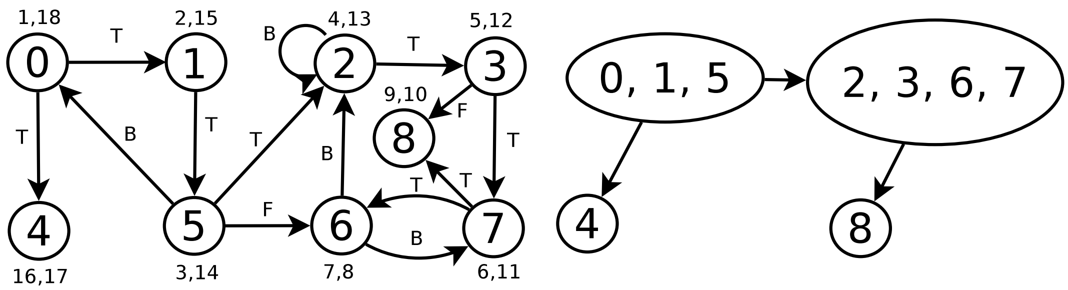
Dat je deo koda za unos grafa sa slike: (rešeno)

```
#define max_cv 50
#include <stdio.h>
#include <stdlib.h>

typedef struct _node gnode;
typedef gnode *grana;
struct _node
{
    int data;
    gnode *next;
};

void enqueue_list(grana **grana_tail_p, int d)
{
    grana grana_new = malloc(sizeof(gnode));

    grana_new -> data = d;
    grana_new -> next = NULL;
    **grana_tail_p = grana_new;
    *grana_tail_p = &(grana_new->next);
}

int main(void)
{
    grana G[max_cv],GT[max_cv];
    int i,n=9; grana *rear[max_cv];

    for(i=0;i<max_cv;i++){
        G[i] = NULL;
         rear[i] = &(G[i]);
    }
    enqueue_list(&rear[0],1); enqueue_list(&rear[0],4);
    enqueue_list(&rear[1],5); enqueue_list(&rear[2],2);
    enqueue_list(&rear[2],3); enqueue_list(&rear[3],7);
    enqueue_list(&rear[3],8); enqueue_list(&rear[5],0);
    enqueue_list(&rear[5],2); enqueue_list(&rear[5],6);
    enqueue_list(&rear[6],2); enqueue_list(&rear[6],7);
    enqueue_list(&rear[7],6); enqueue_list(&rear[7],8);
    return 0;
}
```

5. Napisati kod funkcije enqueue_list koja unosi čvor na kraj liste susedstva grafa. Napisati deo koda za unos grafa sa slike u okviru procedure main u niz listi susedstva G leksikografski.

   Rešeno na prethodnoj strani.

6. Na graf sa slike primeniti DFS algoritam, kod čvorova napisati d i f vrednosti, na grane napisati tip, dati tabelu zagrada. Dati graf komponenti jake povezanosti grafa sa slike.



7. U tabeli su date cene prevoza između 6 gradova.

	1	2	3	4	5	6
1	-	28	31	35	27	18
2	32	-	24	43	45	53
3	23	31	-	54	48	55
4	56	47	55	-	43	25
5	41	46	33	48	-	46
6	60	50	37	29	66	-

   (a) Polazeći od čvora 1, metodom najjeftinijeg suseda naći približno rešenje problema trg. putn. (TSP)

   $1 - 6 - 4 - 5 - 3 - 2 - 1$

   $18 + 29 + 43 + 33 + 31 + 32 = 186$

   (b) Za isti problem naći Mađarskom metodom angažovanje koje je rešenje relaksiranog TSP.

   $1 - 5 - 2 - 3 - 1 \quad 4 - 6 - 4$

   $27 + 46 + 24 + 23 + 25 + 29 = 174$

   (c) Znajući rešenja (a) i (b), naći granice optimalnog rešenja.

   $174 \leq \zeta^* \leq 186$