Fakultet tehničkih nauka, Novi Sad
Animacija u inženjerstvu

17. II 2016. godine

Diskretne i kombinatorne metode za računarsku grafiku

Dat je algoritam

```
1:  function PARTITION(A, p, r)
2:      x ← A[r]
3:      i ← p − 1
4:      for j ← p to r − 1 do
5:          if A[j] ≤ x then
6:              i ← i + 1
7:              exchange(A[i], A[j])
8:          end if
9:      end for
10:     exchange(A[i + 1], A[r])
11:     return i + 1
12: end function
```

1. Posle primene algoritma PARTITION$(A, 1,\ 7)$ na ulaz $A = [5, 1, 8, 2, 9, 6, 3]$, koje će biti stanje niza A?

   $[1, 2, 3, 5, 9, 6, 8]$

2. Koliko poređenja (linija 5) će biti izvršeno na ulaznom nizu iz zadatka 1?

   6

   Koliko puta će se pozvati procedura exchange (linije 7 i 10) za ulazni niz iz zadatka 1 i koliko puta će se zamena u exchange izvršiti?

   3, 3

3. Napisati rekurzivnu proceduru SORT(A, p, r) koja bi korišćenjem procedure PARTITION komandom SORT$(A, 1, 7)$ uradila Quick sort sortiranje niza A.

```
procedure SORT(A, p, r)
    if p < r then
        q ← PARTITION(A, p, r)
        SORT(A, p, q − 1)
        SORT(A, q + 1, r)
    end if
end procedure
```

4. Dati definiciju "velikog O" ponašanja i pokazati da je $n \ln n + n = O(n^2)$.

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

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

Očigledno je $n \ln n + n \geq 0$ za sve $n \in \mathbb{N}$. Pošto je

$$\lim_{n\to\infty} \frac{n \ln n + n}{n^2} = 0\,,$$

sledi da je postoji konstanta $c_1 > 0$ takva da počev od nekog n_0 važi

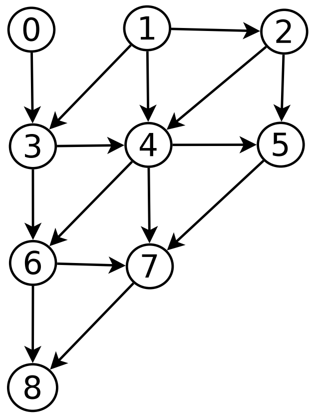
$$\frac{n \ln n + n}{n^2} \leq c_1 \;\Leftrightarrow\; n \ln n + n \leq c_1 n^2\,,$$

što je i trebalo dokazati.

Da li je $n \ln n - n = O(n)$? Ne.

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

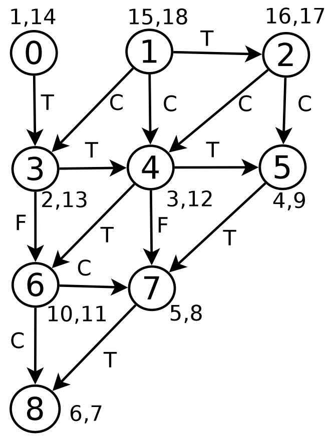
Da li je $\frac{3}{4}n^2 - 3n \ln n = O(n^2)$? Da.

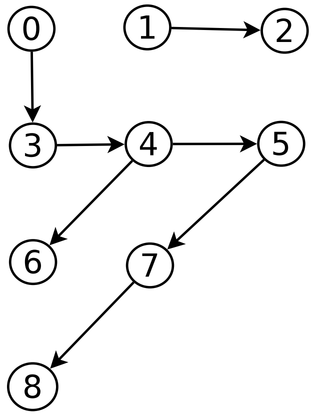


5. Napisati kod funkcije enqueue_list koja unosi čvor na kraj liste susedstva grafa. Napisati deo koda za unos grafa sa slike u okviru procedure main u niz listi susedstva G leksikografski.

```
#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 **g_p, int d)
{
    grana grana_new = malloc(sizeof(gnode));

    grana_new -> data = d;
    grana_new -> next = NULL;
    **g_p = grana_new;
    *g_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],3); enqueue_list(&rear[1],2);
    enqueue_list(&rear[1],3); enqueue_list(&rear[1],4);
    enqueue_list(&rear[2],4); enqueue_list(&rear[2],5);
    enqueue_list(&rear[3],4); enqueue_list(&rear[3],6);
    enqueue_list(&rear[4],5); enqueue_list(&rear[4],6);
    enqueue_list(&rear[4],7); enqueue_list(&rear[5],7);
    enqueue_list(&rear[6],7); enqueue_list(&rear[6],8);
    enqueue_list(&rear[7],8);
    return 0;
}
```

6. Na graf sa slike primeniti DFS algoritam, kod čvorova napisati d i f vrednosti, na grane napisati tip, nacrtati šumu DFS-a. Ako je dati graf usmereni aciklični graf (DAG), dati topološko sortiranje čvorova datog grafa.





Jeste DAG. Topološko sortiranje: 1, 2, 0, 3, 4, 6, 5, 7, 8.

7. U tabeli su date cene prevoza između 5 gradova.
   (a) Polazeći od čvora 1, metodom najjeftinijeg suseda naći približno rešenje probl. trg. putn. (TSP).
   (b) Za isti problem naći Mađarskom metodom angažovanje koje je rešenje relaksiranog TSP.
   (c) Znajući rešenja (a) i (b), naći granice optimalnog rešenja.

	1	2	3	4	5
1	-	8	12	13	4
2	8	-	4	9	9
3	14	5	-	10	6
4	12	8	12	-	7
5	5	9	7	6	-

(a) $\begin{array}{ccccccccccccc} 1 & - & 5 & - & 4 & - & 2 & - & 3 & - & 1 & & \\ & 4 & + & 6 & + & 8 & + & 4 & + & 14 & = & 36 \end{array}$

(b) $\begin{array}{cccccccccccccc} 1 & - & 5 & - & 4 & - & 1 & & 2 & - & 3 & - & 2 & \\ & 4 & + & 6 & + & 12 & & + & & 4 & + & 5 & = & 31 \end{array}$

(c) $31 \leq \zeta \leq 36$.

Uzgred: Optimalno rešenje je: $\begin{array}{ccccccccccccc} 1 & - & 2 & - & 3 & - & 4 & - & 5 & - & 1 & & \\ & 8 & + & 4 & + & 10 & + & 7 & + & 5 & = & 34 \end{array}$