Fakultet tehničkih nauka, Novi Sad
Animacija u inženjerstvu

29. I 2019. godine

Diskr. i k. m. za r. g.

1→15, 2→5, 3→10, 4→10, 5→5, 6→15, 7→10, 8→10.

1. Napisati algoritam za sortiranje biranjem, takozvani SELECTION SORT.

procedure SELECTION SORT(A)
  $n \leftarrow \text{length}(A)$
  for $i \leftarrow 1$ **to** $n-1$ **do**
    $i_{\min} \leftarrow i$
    for $j \leftarrow i+1$ **to** n **do**
      if $A[j] < A[i_{\min}]$ **then**
        $i_{\min} \leftarrow j$
      end if
    end for
    $\text{exchange}(A[i], A[i_{\min}])$
  end for
end procedure

Za algoritam SELECTION SORT iz zadatka 1, za niz dužine n, neka je $S(n)$ broj zamena i $P(n)$ broj poređenja.

2. Za niz $[6, 1, 2, 3, 4, 5]$ naći $S(n)$ i $P(n)$.

   $S(n) = 5,\ P(n) = 15$

3. Koliko je $S(n)$ i $P(n)$ za obrnuto sortirani ulazni niz dužine n algoritma SELECTION SORT?

   $S(n) = \lfloor n/2 \rfloor,\ P(n) = n(n-1)/2$

4. Dati definiciju "malog o" ponašanja i pokazati da je $\frac{1}{4}\, n\, \ln n + 40\, n - 10 = o(n^2)$.

$$o(g) = \{f | (\forall c > 0)(\exists n_0 \in \mathbb{N})(\forall n)\ (n \geq n_0) \Rightarrow (0 \leq f(n) < cg(n)\}$$

Za (počev od nekog n_0) nenegativne funkcije f i g važi

$$f = o(g) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0.$$

Leva strana je nenegativna za $n \geq 1 =: n_0$. Važi:

$$\lim_{n\to\infty} \frac{\frac{1}{4}\, n\, \ln n + 40\, n - 10}{n^2} = \lim_{n\to\infty} \left(\frac{\ln n}{4n} + \frac{40}{n} - \frac{10}{n^2} \right) = 0,$$

odakle sledi da je $\frac{1}{4}\, n\, \ln n + 40\, n - 10 = o(n^2)$

Za niz dužine n neka je $T_{WM}(n)$ najgori slučaj vremena sortiranja Merge sort algoritmom i $T_{WS}(n)$ najgori slučaj vremena sortiranja Selection sort algoritmom. Da li je $T_{WM}(n) = o(T_{WS})$? Da.

Da li je $\frac{3}{4}n^2 + 3n\sqrt{n} = o(n^2 \ln n)$? Da.

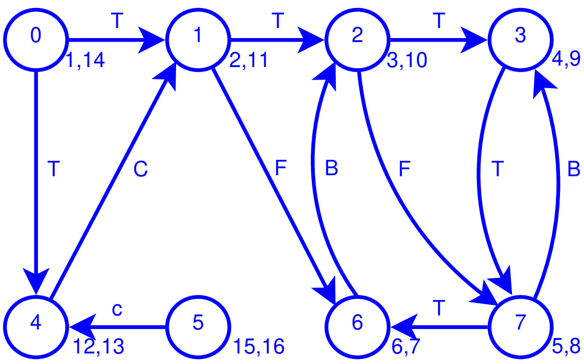
---

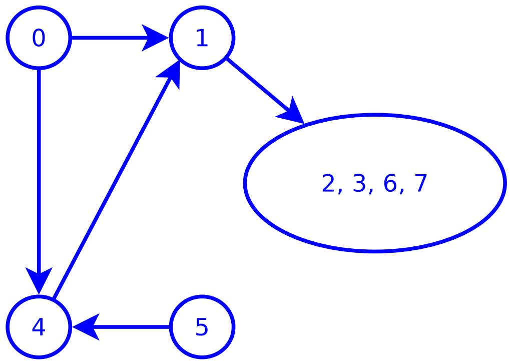
5. Nacrtati usmereni graf G koji je dat tabelom listi susedstva.

6. Na graf G primeniti DFS algoritam, kod čvorova napisati d i f vrednosti, kod grana napisati tip (TBCF), napraviti tabelu zagrada. Ako je dati graf usmereni aciklični graf (DAG), dati topološko sortiranje čvorova, ako nije, dati graf komponenti jake povezanosti.

u	Adj	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	1, 4	(													)		
1	2, 6		(									)					
2	3, 7			(							)						
3	7				(					)							
4	1												(	)			
5	4															(	)
6	2						(	)									
7	3, 6					(			)								

Ovo nije DAG, dole desno je graf komponenti jake povezanosti.





7. Napisati kod funkcije enqueue_list koja unosi čvor na kraj liste susedstva grafa. Napisati deo koda za unos grafa G (iz zad. 5) u okviru procedure main u niz listi susedstva grafa G leksikografski.

```
#define max_cv 50
typedef struct _node gnode;
typedef gnode *grana;
struct _node
{    int data;
     gnode *next;};
```

```
void enqueue_list(grana **grana_tail_p, nextnode d){
    grana grana_new = malloc(sizeof(grana));

    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];
    int i,n; 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],2); enqueue_list(&rear[1],6);
    enqueue_list(&rear[2],3); enqueue_list(&rear[2],7);
    enqueue_list(&rear[3],7); enqueue_list(&rear[4],1);
    enqueue_list(&rear[5],4); enqueue_list(&rear[6],2);
    enqueue_list(&rear[7],3); enqueue_list(&rear[7],6);
    n = 8; return 0;
}
```

8. U tabeli su date cene prevoza između 5 gradova.

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

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

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

(a)

$$1 \to 2 \to 3 \to 5 \to 4 \to 1$$
$$7 + 4 + 6 + 6 + 12 = 35$$

(b)

	1	2	3	4	5
1	-	0	5	6	1
2	4	-	0	5	5
3	9	0	-	5	1
4	5	1	5	-	0
5	0	4	2	1	-
	0	0	0	1	0

	1	2	3	4	5
1	-	0	5	5	1
2	4	-	0	4	5
3	9	0	-	4	1
4	5	1	5	-	0
5	0	4	2	0	-

$k = 4 < 5 \Rightarrow m = 1$

	1	2	3	4	5
1	-	0	4	4	0
2	4	-	0	4	5
3	8	0	-	3	0
4	5	2	5	-	0
5	0	5	2	0	-

$k = 4 < 5 \Rightarrow m = 3$

	1	2	3	4	5
1	-	$\underline{0}$	1	1	0
2	4	-	$\underline{0}$	4	8
3	5	0	-	$\underline{0}$	0
4	2	2	2	-	$\underline{0}$
5	$\underline{0}$	8	2	0	-

Dobili smo angažovanje: $1 \to 2 \to 3 \to 4 \to 5 \to 1$

$$7 + 4 + 10 + 7 + 5 = 33$$

Dobijeno angažovanje pod (b) nema zatvorene potputeve, sledi da je optimalno, $\zeta^* = 33$.
Dobijeno angažovanje pod (a) je dopustivo i nije optimalno, $35 > \zeta^*$.