Fakultet tehničkih nauka, Novi Sad
Animacija u inženjerstvu

11. II 2019. godine

Diskr. i komb. met. za rač. gr.

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

1. Napisati algoritam za sortiranje umetanjem, takozvani INSERTION SORT.

1: **procedure** INSERTION SORT(A)
2: **for** $j \leftarrow 2$ **to** length(A) **do**
3: $key \leftarrow A[j]$
4: $i \leftarrow j - 1$
5: **while** $i > 0\ \&\ A[i] > key$ **do**
6: $A[i+1] \leftarrow A[i]$
7: $i \leftarrow i - 1$
8: **end while**
9: $A[i+1] \leftarrow key$
10: **end for**
11: **end procedure**

Za algoritam INSERTION SORT iz zadatka 1, za niz dužine n, neka je $S(n)$ broj upisivanja elemenata u niz i $P(n)$ broj poređenja.

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

   $S(n) = 10,\ P(n) = 9$

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

   Upisanje elemenata u niz se vrši u linijama 6 i 9, poređenje u liniji 5 ako je $i > 0$.

   $S(n) = 2 + 3 + \cdots + n = \frac{1}{2}(n+2)(n-1)$,

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

4. Dati definiciju "velikog O" ponašanja i pokazati da je $\frac{1}{4} n \ln n + 40 n - 10 = O(n\sqrt{n})$

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

   Za $n \geq 1 =: n_0$ važi da je $\frac{1}{4} n \ln n + 40 n - 10 \geq 0$ i

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

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

   Za niz dužine n neka je $T_{WM}(n)$ najgori slučaj vremena sortiranja Merge sort algoritmom i $T_{BI}(n)$ najbolji slučaj vremena sortiranja Insertion sort algoritmom.

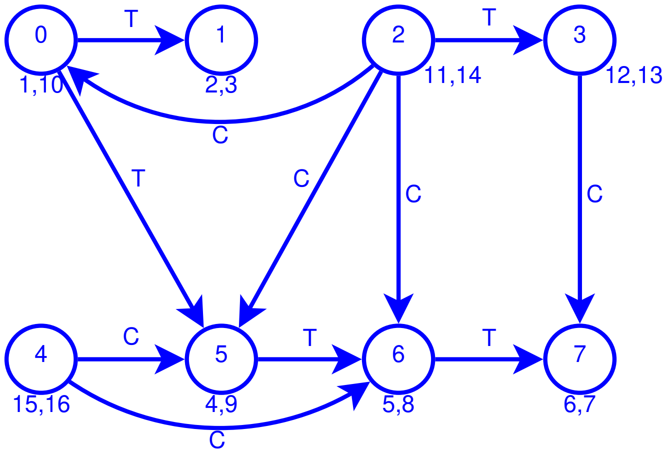
   Da li je $T_{WM}(n) = O(T_{BI})$? Ne.

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

---

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

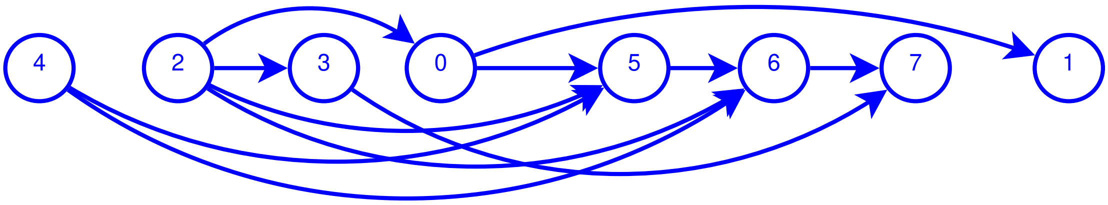
u	Adj(u)
0	1, 5
1	
2	0, 3, 5, 6
3	7
4	5, 6
5	6
6	7
7	



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	$1, 5$	(									)						
1			(	)													
2	$0, 3, 5, 6$											(			)		
3	7												(	)			
4	$5, 6$															(	)
5	6				(					)							
6	7					(			)								
7							(	)									

Ovo je DAG, topološko sortiranje dobijeno primenom DFSa je: 4, 2, 3, 0, 5, 6, 7, 1.



7. Napisati kod funkcije `adjlist2adjmatrix` koja za graf G sa n čvorova dat nizom povezanih listi susedstava nalazi matricu susedstva M. Dati tabelu listi susedstava i matricu susedstva grafa iz zadatka 5.

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

void adjlist2adjmatrix(grana G[], int n, unsigned char M[])
{
    int i;
    grana gr;

    for(i=0;i<n*n;i++) M[i] = 0;
    for(i=0;i<n;i++){
        gr = G[i];
        while(gr){
            M[i*n+gr->data] = 1;
             gr = gr->next;
        }
    }
}

int main(void)
{   grana G[max_cv];
    int i,n; grana *rear[max_cv];
    unsigned char M[max_cv*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],5);
    enqueue_list(&rear[2],0); enqueue_list(&rear[2],3);
    enqueue_list(&rear[2],5); enqueue_list(&rear[2],6);
    enqueue_list(&rear[3],7); enqueue_list(&rear[4],5);
    enqueue_list(&rear[4],6); enqueue_list(&rear[5],6);
    enqueue_list(&rear[6],7); n = 8;
    adjlist2adjmatrix(G, n, M);
    printmatrix(M,n,n);
}
```

Matrica susedstva je:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

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

(a)

$$\begin{array}{ccccccccccc} 1 & \rightarrow & 5 & \rightarrow & 4 & \rightarrow & 3 & \rightarrow & 2 & \rightarrow & 1 \\ & 4 & + & 8 & + & 8 & + & 6 & + & 12 & = & 38 \end{array}$$

(b)

$$\begin{array}{ccccccccccc} 1 & \rightarrow & 2 & \rightarrow & 3 & \rightarrow & 4 & \rightarrow & 5 & \rightarrow & 1 \\ & 10 & + & 7 & + & 5 & + & 7 & + & 5 & = & 34 \end{array}$$

Dobijeno angažovanje pod (b) nema zatvorene potputeve, sledi da je optimalno, $\zeta^* = 34$.

Dobijeno angažovanje pod (a) je dopustivo i nije optimalno, $38 > \zeta^*$.