

Diskretne i kombinatorne metode za računarsku grafiku

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
```

2. Za niz dužine n , neka je $C(n)$ broj poređenja i neka je $W(n)$ broj upisivanja elemenata u niz plus broj upisivanja u privremenu promenljivu.

Za $[5, 2, 4, 6, 1, 3]$ naći $C(n)$ i $W(n)$.

Poređenje se vrši u liniji 5, a upisivanja u linijama 3, 6 i 9.

$$C(n) = 12, W(n) = 19$$

3. Naći $C(n)$ i $W(n)$ iz prethodnog zadatka za niz koji je

(a) sortiran (b) obrnuto sortiran.

(a) $C(n) = n - 1, W(n) = 2(n - 1)$

(b) $C(n) = \frac{n(n-1)}{2}, W(n) = \frac{n(n-1)}{2} + 2(n - 1) = \frac{1}{2}n^2 + \frac{3}{2}n - 2$

4. Dati definiciju "velikog O " ponašanja i pokazati da je $C(n)$ iz zadatka 2 reda $O(n^2)$.

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

Koristimo prethodni zadatak i razumemo da je obrnuto sortiran niz najgori slučaj za broj poređenja: $0 \leq C(n) \leq \frac{n(n-1)}{2}$

Ako nejednakost $0 \leq \frac{n(n-1)}{2} \leq cn^2$ podelimo sa n^2 dobijamo

$$0 \leq \frac{1}{2} - \frac{1}{2n} \leq c.$$

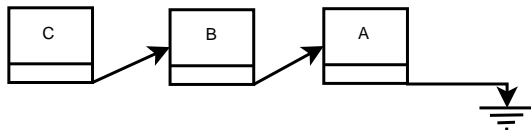
Kako je za $n \geq 1 =: n_0$

$$0 \leq \frac{1}{2} - \frac{1}{2n} \leq \frac{1}{2},$$

možemo uzeti $c := \frac{1}{2}$ i time je pokazano da je

$$\forall n, n \geq n_0 \Rightarrow 0 \leq C(n) \leq cn^2.$$

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 tip podataka cvor:



```
typedef struct _cvor cvor;
struct _cvor
{
    char podatak;
    cvor *sledeci;
};
```

```
int main()
{
    cvor *gomila = NULL;
    cvor *S;

    S = malloc(sizeof(cvor));
    (*S).podatak = 'A';
    (*S).sledeci = NULL;

    gomila = S;

    S = malloc(sizeof(cvor));
    S->podatak = 'B';
    S->sledeci = gomila;
    gomila = S;

    S = malloc(sizeof(cvor));
    S->podatak = 'C';
    S->sledeci = gomila;
    gomila = S;

    while(S)
    {
        printf("%c\n", S->podatak);
        S = S->sledeci;
    }

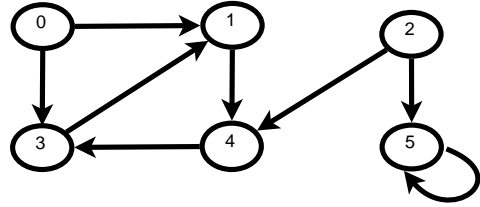
    while(gomila){
        S = gomila;
        gomila = gomila->sledeci;
        free(S);
    }

    return 0;
}
```

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

Napisati proceduru `int stepen(cvor *G[], int i)` koja koristeći reprezentaciju listom susedstva nalazi stepen za čvor i .

U proceduri `stepen` pretpostaviti da je graf zadat nizom pokazivača `G[]` na povezane



i	$Adj(i)$
0	1 3
1	4
2	4 5
3	1
4	3
5	5

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

7. 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.

Držati se leksikografskog redosleda. Napraviti tabelu zagrada. Nacrtati šumu ovog DFS.

Na sledećoj slici vidimo levo: d i f vrednosti i oznaku tipa grane i desno: DFS šumu.

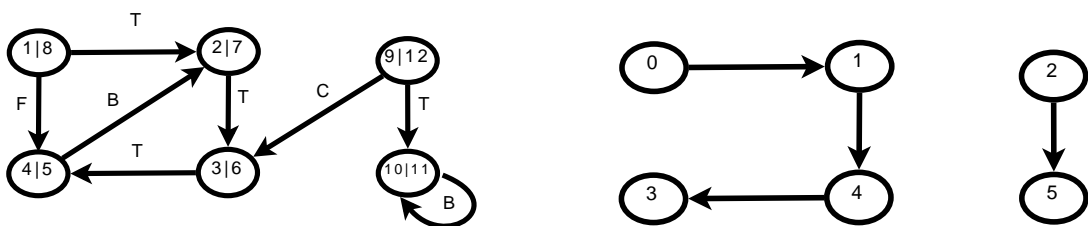


Tabela zagrada:

	1	2	3	4	5	6	7	8	9	10	11	12
0	()				
1		()					
2			(()	
3				()							
4			()							
5									()		

8. Softverska kompanija je zaposlila 5 pripravnika (A, B, C, D, E). Pripravnici će biti angažovani u 5 departmana kompanije (1, 2, 3, 4, 5). Da bi odredili koji pripravnik će se angažovati u kojem departmanu, uradili su test iz veština koje se koriste u odgovarajućem departmanu.

	1	2	3	4	5
A	121	160	130	115	124
B	132	162	140	125	128
C	118	150	142	122	120
D	110	148	129	117	115
E	130	155	135	120	118

U tabeli su dati poeni osvojeni na testu.

Naći optimalno angažovanje.

Od maksimalnog broja bodova 162, ostvarenog kod kandidata B na testu broj 2 oduzimamo ostvarene poene.

	1	2	3	4	5
A	41	2	32	47	38
B	30	0	22	37	34
C	44	12	20	40	42
D	52	14	33	45	47
E	32	7	27	42	44

Optimalno angažovanje dobijamo primenom mađarske metode na dobijenu tabelu.

Rešenje dobijeno mađarskom metodom (međukoraci preskočeni) je naznačeno u matrici:

$$\begin{bmatrix} 41 & \underline{2} & 32 & 47 & 38 \\ 30 & 0 & 22 & 37 & \underline{34} \\ 44 & 12 & \underline{20} & 40 & 42 \\ 52 & 14 & 33 & \underline{45} & 47 \\ \underline{32} & 7 & 27 & 42 & 44 \end{bmatrix}$$

Kada se to rešenje prenese na početnu tabelu dobija se maksimalno angažovanje:

$$160 + 128 + 142 + 117 + 130 = 677.$$