

Diskretne i kombinatorne metode za računarsku grafiku

```

procedure MERGE( $A, p, q, r$ )
  for  $k \leftarrow p$  to  $q$  do
     $L[k - p + 1] \leftarrow A[k]$ 
  end for
   $L[q - p + 2] \leftarrow \infty$ 
  for  $k \leftarrow q + 1$  to  $r$  do
     $R[k - q] \leftarrow A[k]$ 
  end for
   $R[r - q + 1] \leftarrow \infty$ 
   $i \leftarrow 1; j \leftarrow 1$ 
  for  $k \leftarrow p$  to  $r$  do
    if  $L[i] \leq R[j]$  then
       $A[k] \leftarrow L[i]; i \leftarrow i + 1$ 
    else
       $A[k] \leftarrow R[j]; j \leftarrow j + 1$ 
    end if
  end for
end procedure

```

1. Posle primene algoritma MERGE($A, 1, 3, 5$) na ulaz $A = [1, 8, 9, 3, 5, 6, 2, 7, 4]$, koje će biti stanje niza A ?

2. Napisati rekurzivnu proceduru SORT(A, p, r) koja bi korišćenjem MERGE SORT(A) uradila sortiranje niza A .

```

procedure SORT( $A, p, r$ )
  ▷ Ovaj kod napisati
end procedure
procedure MERGE SORT( $A$ )
  SORT( $A, 1, \text{length}(A)$ )
end procedure

```

3. Koliko puta će se pozvati procedura MERGE u toku sortiranja niza $A = [1, 8, 9, 3, 5, 6, 2, 7, 4]$ pozivom MERGE SORT(A)?

4. Dati definiciju "velikog Θ " ponašanja i pokazati da je $n^2 + 3n - 11 = \Theta(n^2)$.

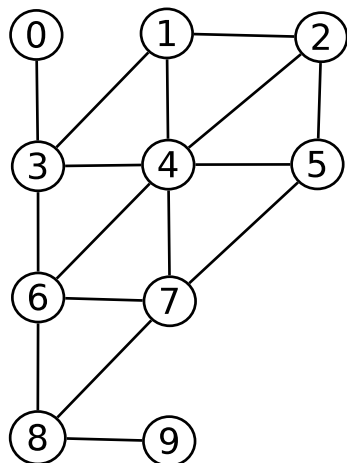
Da li je $n \ln n + n^2 = \Theta(n)$?

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

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

5. Za ADT Queue:

U programskom jeziku C napisati kod funkcije enqueue koja unosi čvor na kraj povezane liste i kod funkcije dequeue koja skida i vraća prvi elemenat povezane liste koja predstavlja queue.



6. Na graf sa slike primeniti BFS algoritam polazeći od čvora 0. Dati tabelu d i p vrednosti. Dati definiciju diametra grafa i naći diameter grafa sa slike.

7. 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.
(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	-

```

typedef struct _node node;
typedef struct _queue *queue;
struct _node
{
  char data;
  node *next;
};
struct _queue
{
  node *front;
  node **rear;
};
typedef struct _queue queues;

int enqueue(queue Q, char d)
{
  node *N_new =
    malloc(sizeof(node));
  if (!N_new) return (1);
  // Ovaj kod napisati
}

char dequeue(queue Q)
{
  // I ovaj kod napisati
}

```

Bodovi: 1→5, 2→15, 3→10, 4→10, 5→15, 6→20, 7→15.