

Zadaci

1. Implementirati *Insertion-sort* algoritam, proveriti njegovu funkcionalnost i analizirati vreme izvršenja. Pseudokod algoritma je prikazan na slici 1.

Insertion-sort funkcija:

```
INSERTION-SORT(A)
1  for j = 2 to A.length
2      key = A[j]
3      // Insert A[j] into the sorted sequence A[1 .. j - 1].
4      i = j - 1
5      while i > 0 and A[i] > key
6          A[i + 1] = A[i]
7          i = i - 1
8      A[i + 1] = key
```

Slika 1 – Pseudokod *Insertion-sort* algoritma

2. Implementirati *Megre-sort* algoritam, proveriti njegovu funkcionalnost i analizirati vreme izvršenja. Pseudokod algoritma je prikazan na slici 2.

Merge-sort funkcija:

```
MERGE-SORT(A, p, r)
1  if p < r
2      q =  $\lfloor (p + r) / 2 \rfloor$ 
3      MERGE-SORT(A, p, q)
4      MERGE-SORT(A, q + 1, r)
5      MERGE(A, p, q, r)
```

Merge funkcija:

```
MERGE(A, p, q, r)
1  n1 = q - p + 1
2  n2 = r - q
3  let L[1 .. n1 + 1] and R[1 .. n2 + 1] be new arrays
4  for i = 1 to n1
5      L[i] = A[p + i - 1]
6  for j = 1 to n2
7      R[j] = A[q + j]
8  L[n1 + 1] = ∞
9  R[n2 + 1] = ∞
10 i = 1
11 j = 1
12 for k = p to r
13     if L[i] ≤ R[j]
14         A[k] = L[i]
15         i = i + 1
16     else A[k] = R[j]
17         j = j + 1
```

Slika 2 – Pseudokod *Merge-sort* algoritma

3. Implementirati algoritam za *linearnu pretragu* brojeva. Ulazni podaci su niz (lista) brojeva i tražena vrednost. Izlaz je pozicija na kojoj se nalazi tražena vrednost.

4. Implementirati algoritam za *binarnu pretragu* brojeva. Ulazni podaci su niz (lista) brojeva i tražena vrednost. Izlaz je pozicija na kojoj se nalazi tražena vrednost.

Napomene:

- Ulazni podaci su celobrojne vrednosti organizovane u listu.
- Funkcionalnost algoritma proveriti na malom broju ulaznih podataka.
- Tokom analize vremena izvršenja algoritma koristiti različite veličine ulaznih podataka.