

## Zadaci

1. Implementirati *Quicksort* algoritam, proveriti njegovu funkcionalnost i analizirati vreme izvršenja. Pseudokodovi algoritma i pomoćnih funkcija su prikazani na slici 1.

<b>PARTITION</b> ( $A, p, r$ ) 1 $x = A[r]$ 2 $i = p - 1$ 3 <b>for</b> $j = p$ <b>to</b> $r - 1$ 4 <b>if</b> $A[j] \leq x$ 5 $i = i + 1$ 6         exchange $A[i]$ with $A[j]$ 7 exchange $A[i + 1]$ with $A[r]$ 8 <b>return</b> $i + 1$	<b>RANDOMIZED-PARTITION</b> ( $A, p, r$ ) 1 $i = \text{RANDOM}(p, r)$ 2 exchange $A[r]$ with $A[i]$ 3 <b>return</b> <b>PARTITION</b> ( $A, p, r$ )  <b>RANDOMIZED-QUICKSORT</b> ( $A, p, r$ ) 1 <b>if</b> $p < r$ 2 $q = \text{RANDOMIZED-PARTITION}(A, p, r)$ 3 <b>RANDOMIZED-QUICKSORT</b> ( $A, p, q - 1$ ) 4 <b>RANDOMIZED-QUICKSORT</b> ( $A, q + 1, r$ )
--	---

Slika 1 – Pseudokodovi *Quicksort* algoritma i pomoćnih funkcija

2. Implementirati *Bucket-sort* algoritam, proveriti njegovu funkcionalnost i analizirati vreme izvršenja. Pseudokodovi algoritma i pomoćnih funkcija su prikazani na slici 2.

<b>BUCKET-SORT</b> ( $A$ ) 1 let $B[0 \dots n - 1]$ be a new array 2 $n = A.length$ 3 <b>for</b> $i = 0$ <b>to</b> $n - 1$ 4     make $B[i]$ an empty list 5 <b>for</b> $i = 1$ <b>to</b> $n$ 6     insert $A[i]$ into list $B[\lfloor nA[i] \rfloor]$ 7 <b>for</b> $i = 0$ <b>to</b> $n - 1$ 8     sort list $B[i]$ with insertion sort 9 concatenate the lists $B[0], B[1], \dots, B[n - 1]$ together in order
---

Slika 2 – Pseudokod *Bucket-sort* algoritma

3. Implementirati *Counting-sort* algoritam, proveriti njegovu funkcionalnost i analizirati vreme izvršenja. Pseudokodovi algoritma i pomoćnih funkcija su prikazani na slici 3.

```
COUNTING-SORT( $A, B, k$ )
1  let  $C[0..k]$  be a new array
2  for  $i = 0$  to  $k$ 
3       $C[i] = 0$ 
4  for  $j = 1$  to  $A.length$ 
5       $C[A[j]] = C[A[j]] + 1$ 
6  //  $C[i]$  now contains the number of elements equal to  $i$ .
7  for  $i = 1$  to  $k$ 
8       $C[i] = C[i] + C[i - 1]$ 
9  //  $C[i]$  now contains the number of elements less than or equal to  $i$ .
10 for  $j = A.length$  downto 1
11      $B[C[A[j]]] = A[j]$ 
12      $C[A[j]] = C[A[j]] - 1$ 
```

Slika 3 – Pseudokod *Counting-sort* algoritma

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.