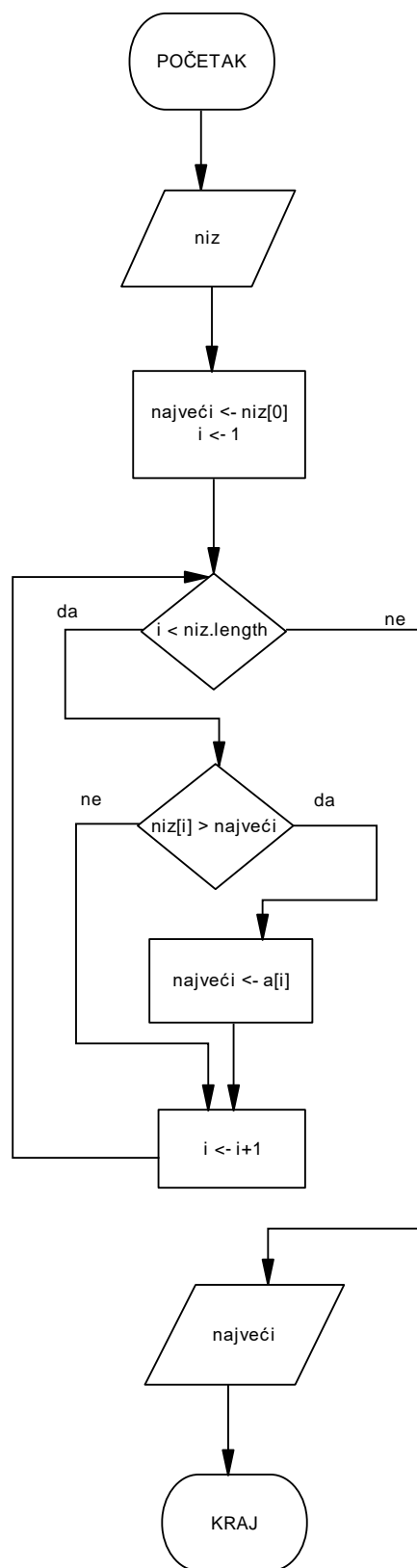


1 – Blok dijagram algoritma dat je slikom ispod



Slika – algoritam pronalaženja najvećeg elementa u nizu

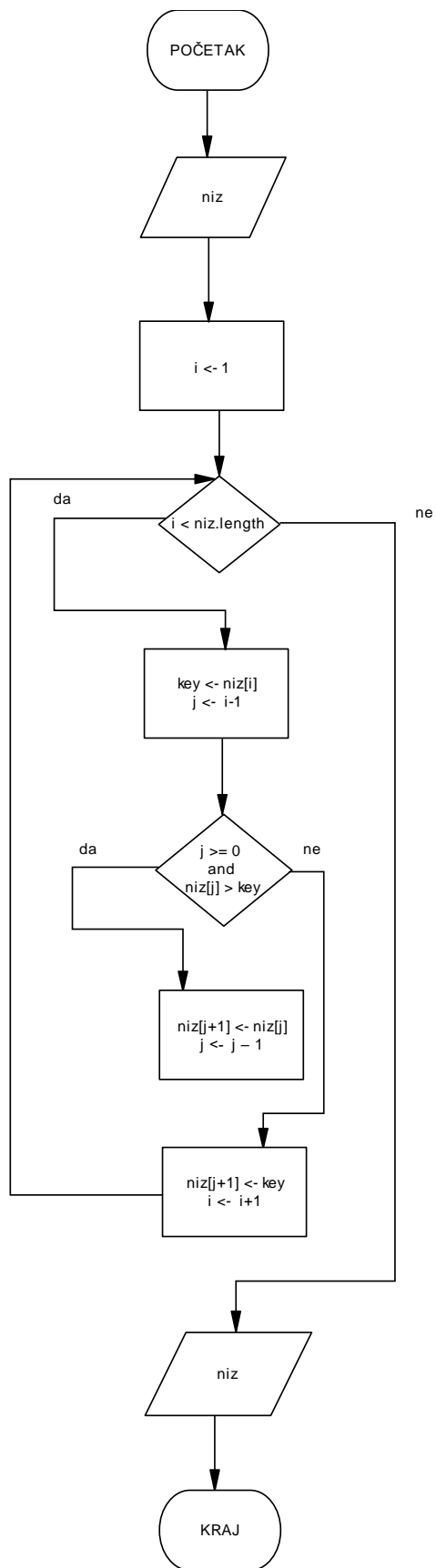
Algoritam pronalaženja najvećeg element u nizu brojeva će pristupiti po jednom svakom elementu niza da bi proverio da li je dati element veći od trenutno najvećeg elementa. Stoga je složenost ovog elementa $\Theta(n)$.

2 – Obzirom da bucket sort algoritam ima samo jednu for petlju, telo te petlje će se izvršiti onoliko puta koliko ima elemenata u ulaznom nizu. Stoga je složenost ovog algoritma $\Theta(n)$.

3 – Ako niz ima n elemenata kroz sve elemente niza prolazimo n puta. U svakom prolazu svaki element poredimo sa elementom koji mu sledi. Ukoliko je element koji sledi manji od trenutnog elementa zamenimo vrednosti u trenutnom i sledećem elementu elementa.

Obzirom da bubble sort algoritam ima jednu ugnježdenu petlju, telo petlje će se izvršavati po n puta za svaki element iz ulaznog niza. Stoga je složenost ovog algoritma $\Theta(n^2)$.

4 – Insertion sort radi tako što niz deli u dva dela: nesortiran i sortiran. U svakom koraku preuzima se prvi element nesortiranog niza i postavlja se na odgovarajuću poziciju u sortiranom delu niza. Znači, za svaki element iz niza, pomere se svi elementi koji mu prethode i koji su manji od njega za po jedno mesto u desno u nizu. Time se oslobodi mesto na koje se postavi trenutni element niza.



Slika – blok dijagram insertion sort algoritma

Obzirom da insertion sort ima jednu ugnježdenu petlju (za svaki element niza potrebno je proći kroz elemente koji mu prethode), telo petlje će se izvršavati po n puta za svaki element iz ulaznog niza. Stoga je složenost ovog algoritma $\Theta(n^2)$.

5 – Poređenje algoritama:

Najefikasniji je bucket sort algoritam, ali je vrlo ograničene primene (može se koristiti samo ako sortiramo niz celih brojeva koji čini sekvenca elemenata).

Preostali analizirani algoritmi su jednaka složenosti i podjednako su primenjivi.