

Primer: zamena vrednosti dve promenljive

```
// Ulaz:  promenljive x i y sa svojim vrednostima
// Izlaz: promenljive x i y sa zamenjenim vrednostima
algorithm swap(x, y)

    z = x;
    x = y;
    y = z;

    return x, y;
```

Primer: indeks najvećeg elementa niza

```
// Ulaz: niz a, broj elemenata n niza a
// Izlaz: indeks najvećeg elementa niza a
algorithm max(a, n)

m = a[1];      // najveći element nadjen do sada
j = 1;        // indeks najvećeg elementa

i = 2;
while (i <= n) do
    if (m < a[i]) then      // nadjen veći element od
                            // privremeno najvećeg
        m = a[i];          // zapamti veći broj
        j = i;              // i njegov indeks
    i = i + 1;             // predji na sledeći element

return j;        // vrati indeks najvećeg elementa
```

ili,

Primer: najveći element niza

```
// Ulaz: niz a, broj elemenata n niza a
// Izlaz: vrednost najvećeg elementa niza a
algorithm max(a, n)

m = a[1];      // najveći element nadjen do sada
j = 1;        // indeks najvećeg elementa

i = 2;
while (i <= n) do
    if (m < a[i]) then      // nadjen veći element od
                            // privremeno najvećeg
        m = a[i];          // zapamti veći broj
        j = i;              // i njegov indeks
    i = i + 1;             // predji na sledeći element

return a[j];        // vrati vrednost najvećeg elementa
```

$$T(n) = 1 + 1 + 1 + (n - 1)(3 + 1) + 1 = 4 + 4(n - 1) = 4n.$$

```
// Ulaz: niz A, broj elemenata n niza A
// Izlaz: maksimalni element niza A
algorithm max(a, n)

m = a[1];      // najveći element nadjen do sada

for i=2 to n do

    if (m < a[i]) then

        m = a[i]; // nadjen veći element od privremeno
                  // najvećeg

return m;      // vrati maksimalni element niza
```

$$T(n) = 1 + (n - 1)(1 + 1) + 1 = 2n$$

Rekurzivni algoritam:

```
// Ulaz: niz A, broj elemenata n niza A
// Izlaz: maksimalni element niza A
algorithm Maximum(A, n)

  if (n == 1) then          // bazični slučaj
    return (A[1]);

  else                      // opšti slučaj

    return (max(Maximum(A,n-1),A[n]));
```

```
// Ulaz: niz a, broj elemenata n niza a
// Izlaz: indeks najmanjeg elementa niza a
algorithm max(a, n)

m = a[1];      // najmanji element nadjen do sada
j = 1;        // indeks najmanjeg elementa

i = 2;
while (i <= n) do
    if (m > a[i]) then      // nadjen manji element od privremeno
                            // najmanjeg
        m = a[i];          // zapamti manji broj
        j = i;             // i njegov indeks
    i = i + 1;            // predji na sledeci element

return j;          // vrati indeks najmanjeg elementa
```