Merge sort

Merge sort je jedan od algoritama koji i u najgorem slučaju garantuje složenost $O(n \log n)$. Međutim, zbog konstante iza ove složenosti, quicksort se češće koristi. Kao i quicksort, merge sort radi po principu podeli pa vladaj.

Ideja merge sort algoritma je da podeli početni niz dužine n, na dva niza jednakih dužina od po $n/2$ elemenata, sortira rekurzivno svaki od ovih nizova i na kraju ih 'spoji' u jedan sortirani niz.

Moguća implementacija mergesort-a je prikazana u *Algoritam 3*, gde funkcija $merge(a, left, mid, right)$ kao ulaz prima dva sortirana niza ($a[left..mid]$ i $a[mid + 1..right]$) i spaja ih u niz a.

```
================================================================================
            funkcija: mergesort
            ulaz:  a      - niz brojeva
                   left   - indeks prvog elementa u delu niza koji posmatramo
                   right  - indeks poslednjeg elementa u delu niza koji posmatramo

            nakon izvršavanja funkcije mergesort(a, left, right), podniz
            a[left..right] će biti sortiran
--------------------------------------------------------------------------------
      Function mergesort(a : int array, left : int, right : int)
             01     if left < right then          // imamo više od jednog elementa
             02         mid = (left + right) / 2 // uzmi srednji element
             03         mergesort(a, left, mid)
             04         mergesort(a, mid + 1, right)
             05         merge(a, left, mid, right)
================================================================================
```

Algoritam 3. Pseudo kod za mergesort

Spajanje dva sortirana niza u jedan se radi uzimanjem prvih elemenata iz oba niza i stavljanja manjeg na kraj konačnog niza i njegovo brisanje iz niza iz kojeg je uzet. Ponavljajući ovaj postupak dokle god imamo bar jedan elemenat u nekom od početna dva niza, dobijamo sortirani niz koji se sastoji od brojeva iz početna dva niza. Implementacija ove ideje je prikazana u *Algoritam 4*.

```
================================================================================
            funkcija: merge
            ulaz:  a      - niz brojeva
                   left   - indeks prvog elementa prvog dela niza koji posmatramo
                   mid    - indeks poslednjeg elementa prvog sortiranog niza
                   right  - indeks poslednjeg elementa drugog dela niza

            nakon izvršavanja funkcije merge(a, left, mid, right) podniz
            a[left..right] će biti sortiran
--------------------------------------------------------------------------------
      Function merge(a : int array, left : int, mid : int, right : int)
             01         b : int array[1..right - left + 1]
             02         position = 1
             03         first = left       // indeks sledećeg elementa u prvom nizu
             04         second = mid + 1   // indeks sledećeg elementa u drugom nizu
             05         For step = left to right do
             06             If (second > right or
             07                 first <= mid and a[first] <= a[second]) then
             08                 // element prvog niza je manji
             09                 b[position] = a[first]
             10                 first = first + 1
             11                 position = position + 1
             12             Else
             13                 // element drugog niza je manji
             14                 b[position] = a[second]
             15                 second = second + 1
             16                 position = position + 1
             17
             18         // sada niz b sadrzi sortiran podniz a[left..right]
             19         // kopirajmo niz b u niz a
             20         For i = left to right do
             21             a[i] = b[i - left + 1]
================================================================================
```

Algoritam 4. Pseudo kod za merge

Primetimo da nam nije potreban niz b duzine $right - left + 1$. Za vežbu se ostavlja da se napiše funkcija merge sa dodatnim nizom duzine $(right - left + 1)/2$.

Koja je složenost merge sort-a? Najpre primetimo da je složenost funkcije $merge$ $O(right - left)$, tj. linearna u odnosu na zbir veličina dva sortirana niza koja treba spojiti.

Označimo složenost merge sort-a prilikom sortiranja niza sa n elemenata sa $f(n)$. Rekurenta relacija je:

$$f(n) = O(n) + 2f(n/2),$$ gde dobijamo da je $f(n) = O(n \log n)$

Ili kao I kod quicksort-a, možemo da primetimo da u svakom koraku rekurzije prolazimo kroz čitav niz, pa je složenost $f(n) = O(n) * DubinaRekurzije$, a kod merge sort-a imamo da je dubina rekurzije $\log n$, pa je samim tim složenost istog $O(n \log n)$.

Ovaj pristup merge sort-u, gde krenemo sa čitavim nizom i rekurzivno delimo na dva niza, se naziva **top-down merge sort**. Alternativa je da krenemo sa n nizova od po jednim elementom i spajamo susedne nizova, pa tako posle ovog koraka dobijamo $n/2$ nizova sa po 2 elementa. Daljom primenom istog dobijamo $n/4$ nizova sa po 4 elementa, itd. Na kraju dobijamo sortirani niz. Ovaj pristup se naziva **bottom-up merge sort**.