

Quicksort – kad je stožer na prvom mjestu

Na početku imamo niz

5 4 9 1 7 10 8 2 3 6

Stožer je prvi član polja (član s indeksom 0) odnosno broj 5.

Cilj prvog koraka Quicksorta je postići da se na lijevoj strani polja nalaze brojevi manji od stožera, a na desnoj strani da se nalaze brojevi veći od stožera.

Pokazivač **i** postavimo na 2. član polja (onaj s indeksom 1) → lijevo+1

5 4 9 1 7 10 8 2 3 6

Pokazivač **j** postavimo na zadnji član polja.

5 4 9 1 7 10 8 2 3 6

```
int i, j;
tip pom;
i = lijevo+1; // pokazivač i ide odmah desno od stožera, na drugi član polja
j = desno;    // pokazivač j ide na kraj polja (zadnji član)
```

Kreće glavni dio Quicksorta:

Pokazivač **i** mičemo prema desno SVE DOK JE MANJA VRIJEDNOST NA KOJU POKAZIVAČ **i** POKAZUJE OD STOŽERA

5 4 9 1 7 10 8 2 3 6

i se nalazi na početku na „broju 4“ pošto je $4 < 5$ ide jedan korak desno.

Kao što vidimo kad **i** dođe na 3.član polja odnosno broj 9, tu će se zaustaviti jer nije ispunjen uvjet za ići dalje odnosno $9 < 5$ (FALSE)

j pokazivač ide prema LIJEVO SVE DOK JE VRIJEDNOST NA KOJU POKAZIVAČ **j** POKAZUJE VEĆA OD STOŽERA

5 4 9 1 7 10 8 2 3 6

j pokazivač će se zaustavit na „broju 3“

```
if (lijevo >= desno) return; // Quicksort ide sve dok je lijeva strana polja
                             manja od desne strane polja

(3) while ((i <= j) && (i<=desno) && (j>lijevo)) { // j može ići samo do 2.člana
    (1) while ((A[i] < A[lijevo]) && (i<=desno)) i++; // dok je < od stožera...
    (2) while ((A[j] > A[lijevo]) && (j>lijevo)) j--; // dok je > od stožera...
    if (i<j) {
        Zamijeni (&A [i], &A [j]);
    }
} // gotova petlja
```

s brojem 1 je označena naredba koja se brine da pokazivač **i** ide desno sve dok je vrijednost na koju pokazuje manja od stožera, a stožer se nalazi na početku polja → `while ((A[i] < A[lijevo])` idi desno odnosno `i++!`

Za **j** vrijedi da ide lijevo dok je vrijednost na koju pokazuje veća od stožera

→ `while ((A[j] > A[lijevo]) idi lijevo → j--`

Kad puknu i jedna i druga while petlja stanje je slijedeće:

5 4 9 1 7 10 8 2 3 6

onda ide provjera da li su se pokazivači `i` i `j` mimoišli → `if (i < j) {`
ako nisu → Zamijeni (`&A [i], &A [j]`);

5 4 3 1 7 10 8 2 9 6

Sad opet idemo kroz glavnu petlju (označenu s brojem 3) jer se `i` i `j` nisu mimoišli.

`while ((i <= j) && (i <= desno) && (j > lijevo))`

Dakle, ponavljamo priču:

5 4 3 1 7 10 8 2 9 6

`i` ide desno dok je manji od stožera:

`j` ide lijevo dok je manji od stožera:

5 4 3 1 7 10 8 2 9 6

nisu se još mimoišli pa slijedi zamjena:

5 4 3 1 2 10 8 7 9 6

još se nisu `i` i `j` mimoišli → idemo dalje

obratite pozornost da smo postigli cilj → prvih 5 članova je manje ili jednako od stožera, drugih 5 je veće od stožera.

ali trebamo čekati da se pokazivači `i` i `j` mimođu

5 4 3 1 2 10 8 7 9 6

idemo:

5 4 3 1 2 10 8 7 9 6

`j`(zeleni) pokazuje na 2 jer je 2 manji od stožera (to ga je tu i zaustavilo)

`i`(plavi) pokazuje na 10

one dvije petlje su ih tako pozicionirale:

```
while ((A[i] < A[lijevo]) && (i <= desno)) i++; // dok je < od stožera...
while ((A[j] > A[lijevo]) && (j > lijevo)) j--; // dok je > od stožera...
```

nakon njih je slijedio:

```
if (i < j) {
    Zamijeni (&A [i], &A [j]);
}
```

uvijek u `if`-u se neće ispuniti jer su se pokazivači mimoišli.

onda se neće ispuniti ni uvijet u vanjskoj petlji...

```
while ((i <= j) && (i<=desno) && (j>lijevo))
```

i sve završava...

```
// sad možemo imati tri moguće situacije
```

```
if (i > desno) { // stožer je najveći u polju, i je išao sve dok je < od...
    Zamijeni (&A [lijevo], &A [desno]);
    Qsort2(A, lijevo, desno-1); // zovi Qsort2 za preostalo polje
}
else if (j<=lijevo) { // stožer je najmanji u polju
    Qsort2(A, lijevo+1, desno); // zovi Qsort2 za preostalo polje
}
else { // stožer je negdje u sredini
    Zamijeni (&A [lijevo], &A [j]);
    Qsort2(A, lijevo, j-1); // zovi Qsort2 za lijevu stranu preostalog
    Qsort2(A, j+1, desno); // zovi Qsort2 za desnu stranu preostalog
}
}
```

Ove prve dvije situacije ću prokomentirati, iako nisu naš slučaj:

1. ako je **i** na poziciji **desno+1** → znači da su svi elementi polja manji od prvog člana odnosno stožera (i ide desno sve dok su vrijednosti na koje pokazuje manje od stožera)

→ dakle, stožer je najveći u polju, stoga ga stavljamo na kraj

→ zovemo opet Quicksort za polje od početka do **desno-1** (ne uključujemo stožer)

u ovom slučaju pokazivač **j** se neće pomaknuti sa svog mjesta :

prvo je išla ona **while** petlja koja miče **i** i ona će **i** dovesti do lokacije **desno +1**

zatim ide petlja koja miče **j**, ali **j** se miče samo ako pokazuje na vrijednost koja je veća od stožera, a pošto je stožer najveći... **j** ostaje na svom mjestu

tako se **i** mimoiđu, jer **j** ostane na kraju polja (**desno**) , a **i** ostane na **desno+1**

2. slučaj je kad se **j** pomakne na poziciju lijevo

→ ako je stožer najmanji , **j** se stalno miče ulijevo jer pokazuje na članove koju su veći od stožera, to radi dok ne dođe do samog stožera, a pošto **i** pokazuje na **lijevo+1** na početku, tako će se **i** mimoići s njim

3. je NAŠ SLUČAJ

5 4 3 1 2 10 8 7 9 6

stožer treba dijeliti polje na dva dijela tako da su lijevo manji, a desno veći

ako zamijenimo stožer s mjestom **na koje pokazuje pokazivač j (zeleno)**

2 4 3 1 5 10 8 7 9 6

```
Zamijeni (&A [lijevo], &A [j]);
Qsort2(A, lijevo, j-1); // zovi Qsort2 za lijevu stranu preostalog
Qsort2(A, j+1, desno); // zovi Qsort2 za desnu stranu preostalog
```

ponavljamo isti postupak za lijevu stranu polja (stožer je prvi član u lijevoj strani polja)

ponavljamo isti postupak za desnu stranu polja (stožer je prvi član u desnoj strani polja)