

Sveučilište u Splitu  
Fakultet elektrotehnike, strojarstva i brodogradnje

Algoritmi

# Vježba 8

Nositelj kolegija: izv.prof.dr.sc Matko Šarić  
Suradnici u nastavi: asistent Marin Maslov, mag.ing.

## Vježba 8

U priloženom predlošku implementirajte funkcije *Partition*, *QuickSort* i *BubbleSort*. Pokrenite *Quick sort*, ispišite vrijeme izvršavanja i spremite sortirane vrijednosti u *txt* datoteku. Isto ponovite i za *Bubble sort*. Za vrijeme izvršavanja koristite funkciju:

```
int GetTickCount(void) //include windows.h;
```

koja vraća vrijeme koje je proteklo od trenutka pokretanja sustava (u milisekundama).

Pokušajte pokrenuti Quick sort i Bubble sort za par nizova različitih veličina, npr. za *size = 100, 1000, 10000, 20000, 30000, 50000*. Zapišite vremena izvršavanja za različite veličine niza. Točnost algoritma provjerite otvaranjem *txt* datoteke u kojoj vrijednosti trebaju biti zapisane u rastućem redoslijedu.

### Pseudokod

```
BubbleSort (array A, int p, int r)
    for i=p to r-1
        for j=i+1 to r
            if(A[i]>A[j])
                Swap(A[i],A[j])
```

```
Partition (int p, int r, array A)
    x = A[p] //x je jednak prvom elementu
    q = p
    for s=p+1 to r do
        if (A[s]<x)
            q=q+1
            swap A[q] with A[s]
    swap A[p] with A[q] //stavi element x na odgovarajuće mjesto
    return q //vrati poziciju elementa x
```

```
QuickSort (int p, int r, array A)
    if (r <= p) return //0 ili jedan element, return
    i = slučajni indeks iz [p..r] //odabiranje slučajnog indeksa (elementa)
    zamijeni A[i] s A[p] //stavljanje x na odgovarajuće mjesto
    q = Partition (p, r, A) //podjela A oko elementa x
    QuickSort (p, q-1, A) // sortiranje A[p..q-1]
    QuickSort (q+1, r, A) //sortiranje A[q+1..r]
```

## Predložak za implementaciju (c/c++)

```
#include <stdio.h>
#include <tchar.h>
#include<stdio.h>
#include<windows.h>
#include<time.h>
#include<conio.h>
#include <math.h>
#include <time.h>
#include<iostream>

using namespace std;

void Swap(int &a, int &b) {
    int temp;
    temp=a;
    a=b;
    b=temp;
}

int Partition(int p, int r, int A[]) {
    //implementiraj partition
}

void QuickSort(int p, int r, int A[]) {
    //implementiraj quick sort
}

void BubbleSort(int size, int A[]) {
    // implementirajte bubble sort
}

int main() {
    srand( (unsigned)time( NULL ) ); //usklađivanje rand generatora sa sistemskim
    vremenom
    FILE *file1,*file2;
    int vrijeme1,vrijeme2;
    int i,size;

    printf("Unesi velicinu niza\n");
    scanf("%d",&size);

    //generiranje niza
    int *A=(int*)malloc(size*sizeof(int));
```

```

    for(i=0;i<size;i++)
        A[i]=rand();

    vrijeme1=GetTickCount();
    //pokreni quick sort

    vrijeme2=GetTickCount();
    // ispiši vrijeme sortiranja
    printf("vrijeme1:%d\n",vrijeme2-vrijeme1);

    //ispis u file
    file1 = fopen( "OutputQuickSort.txt", "w" );
    for(i=0;i<size;i=i++)
        fprintf(file1, "%d\n",A[i]);
    fclose(file1);

    //ponovno generiranje niza, pokretanje bubble sorta, ispis vremena i
    sortiranog niza u txt file

    free(A);
    return 0;
}

```