Računske vježbe 5

Programiranje I

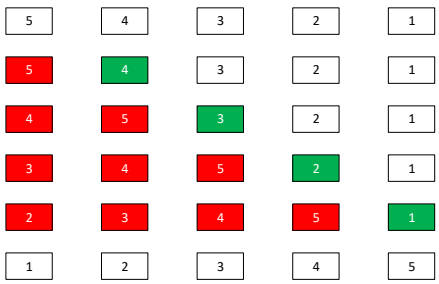
1. Napisati program koji na unaprijed sortiranom nizu (rastući poredak) primjenjuje binarno pretraživanje uz pomoć odgovarajuće rekurzivne funkcije.

```c
#include <stdio.h>

int binary_search(int [], int, int, int);

int main()
{
    int array[30], length, i, position, elem;

    printf("Unesite duzinu niza:\n");
    scanf("%d", &length);

    printf("Unesite elemente niza:\n");
    for(i = 0; i < length; i++)
        scanf("%d", &array[i]);

    printf("Unesite element koji se trazi:\n");
    scanf("%d", &elem);

    position = binary_search(array, 0, length - 1, elem);
    if(position == -1)
        printf("Trazeni element se ne nalazi u datom nizu!");
    else
        printf("Trazeni element se nalazi na poziciji %d.", position);
}

int binary_search(int arr[], int start, int end, int elem)
{
    if (end >= start)
    {
        int mid = start + (end - start) / 2;
        // prvo provjeravamo da li se element nalazi tacno
        //  u sredini niza arr
        if(arr[mid] == elem)
            return mid;
        // ukoliko je element manji od arr[mid] onda je
        // jasno da se moze naci jedino u prvoj polovini
        if (arr[mid] > elem)
            return binary_search(arr, start, mid - 1, elem);
        else
            return binary_search(arr, mid + 1, end, elem);
        // ukoliko nije manji od arr[mid] moze se naci
        // jedino u drugoj polovini
    }
    return -1;
}
```

2. Napisati program kojim se unosi niz cijelih brojeva, dužine N, koji sortira niz u rastući redosljed, koristeći algoritam **Insertion sort**.

```c
#include <stdio.h>

int main()
{
    int array[30], n, i, j, curr_elem;

    printf("Unesite duzinu niza:\n");
    scanf("%d", &n);

    printf("Unesite elemente niza:\n");
    for(i = 0; i < n; i++)
        scanf("%d", &array[i]);

    for(i = 1; i < n; i++)
    {
        curr_elem = array[i];
        j = i - 1;
        while(j >= 0 && array[j] > curr_elem)
        {
            array[j + 1] = array[j];
            j--;
        }
        array[j + 1] = curr_elem;
    }

    printf("Sortirani niz je:\n");
    for(i = 0; i < n; i++)
        printf("%d ", array[i]);
}
```

Ovaj jednostavni algoritam sortiranja funkcioniše na sledeći način. Niz se dijeli na dva dijela, sortirani i nesortirani. U početku samo prvi element niza pripada sortiranom, a svi ostali nesortiranom dijelu. Uzima se prvi element iz nesortiranog dijela i umeće (insertion) u sortirani dio na odgovarajuće mjesto što se ponavlja za sve elemente nesortiranog dijela. Slijedi kratko objašnjenje koda. For petljom prolazimo kroz niz od drugog do poslednjeg elementa. Ideja je da uvijek gledamo iza tekućeg elementa znajući da se iza njega nalazi sortirani dio niza. Za dato `i` imamo tekući element `array[i]`. Sada posmatramo sve elemente u intervalu `0..i-1` i pomjeramo ih za jednu poziciju udesno ukoliko su veći od tekućeg elementa kako bismo napravili prostor za njega. To radimo sa brojačem `j` koji kreće sa pozicije `i - 1` i smanjuje se sve dok iza tekućeg elementa ima onih koji su od njega veći. Kada ovaj uslov više nije zadovoljen preostaje nam da samo smjestimo tekući element.



Slika 1: Zelenom bojom označen je tekući element (`array[i]`) dok su crvenom bojom označeni elementi u intervalu `0..i-1` koji su veći od tekućeg odnosno oni koje je za jednu poziciju potrebno pomjeriti udesno.