

Algoritmi i strukture podataka – međuispit

26. travnja 2016.

Nije dopušteno korištenje globalnih i statičkih varijabli te naredbe goto. Ispit donosi maksimalno 30 bodova. Ovaj primjerak ispita trebate predati s upisanim imenom i prezimenom te JMBAG-om.

Zadatak 1. (6 bodova)

Napišite rekurzivnu funkciju prototipa:

int PascalovTrokut (int redak, int stupac);

koja određuje i vraća cijeli broj koji se nalazi u retku **redak** i stupcu **stupac** u Pascalovom trokutu. Prikaz Pascalovog trokuta u obliku matrice:

	0	1	2	3	4
0	1				
1	1	1			
2	1	2	1		
3	1	3	3	1	
4	1	4	6	4	1

itd.

Primjeri:

- za poziv PascalovTrokut(0, 0), funkcija treba vratiti 1
- za poziv PascalovTrokut(1, 1), funkcija treba vratiti 1
- za poziv PascalovTrokut(2, 1), funkcija treba vratiti 2
- za poziv PascalovTrokut(4, 2), funkcija treba vratiti 6 (primjer sa slike)

$6 = 3 + 3$

Napomena: nerekurzivno rješenje se neće priznati.

Zadatak 2. (6 bodova)

Odredite vrijeme izvođenja u O , Ω i, ako je moguće, Θ notaciji za programski odsječak u a) dijelu zadatka i za funkciju **rekurzija** u b) dijelu zadatka. Ako se vrijeme izvođenja u Θ notaciji ne može odrediti, navedite tako u rješenju. Rješenja upišite u tablice pored zadatka.

a)

```
/* A je polje n cijelih brojeva */
if (n <= 10) {
    g(n); /* obavlja se u  $\Theta(n)$  vremenu */
}
else {
    for (i = n - 1; i > 0; i--) {
        if (A[i] > A[i - 1]) {
            g(i);
        }
    }
}
```

O	
Ω	
Θ	

b)

```
void rekurzija(int n) {
    int i;
    if (n == 0) {
        return;
    }
    for (i = 0; i < n; i++) {
        rekurzija(i);
    }
}
```

O	
Ω	
Θ	

Zadatak 3. (6 bodova)

Za zadani program prikažite sadržaj sistemskog stoga počevši s pozivom funkcije **f1** iz glavnog programa (linija 20) te sve do neposredno prije izvođenja naredbe **return 0**; u funkciji **f2** (linija 8). Uz svaku stavku napišite broj okteta (byte-ova) koje stavka zauzima na stogu.

Sadržaj stoga te broj okteta koje zauzima svaka stavka prikažite u pravokutniku desno od zadanog programa.

```
1  #include <stdio.h>
2  int f2(int *a, int *b) {
3      if (*a > *b) {
4          --(*a);
5          f2(a, b);
6      }
7      else {
8          return 0;
9      }
10 }
11
12 void f1(int *a, int *b) {
13     int *c = a, *d = b, t;
14     printf("%d %d", *c, *d);
15     t = f2(a, b);
16 }
17
18 int main(void) {
19     int A[] = {4, 3, 2, 1};
20     f1(&A[0], &A[1]);
21     return 0;
22 }
```

Sadržaj stoga:

--

Zadatak 4. (6 bodova)

Zadano je polje brojeva s elementima: **8, 7, 2, 1, 4, 6, 8, 1**. Ilustrirajte uzlazno sortiranje zadanog niza brojeva (ispišite polje nakon svake promjene i podcrtajte sve brojeve relevantne za sljedeći korak) algoritmom **shellsort** uz korake **k = {4, 3, 1}**.

Zadatak 5. (6 bodova)

Napišite funkciju `stvari_matricu` koja stvara odredišnu cjelobrojnu matricu veličine $n \times n$ temeljem ulaznog polja cijelih brojeva veličine n . Svaki redak odredišne matrice jednak je ulaznom polju, uz iznimku da elementi dijagonale matrice trebaju biti 0.

Potrebno je napisati i glavni program koji stvara polje od 4 elementa s vrijednostima {1, 5, 3, 7} te poziva funkciju `stvari_matricu`. Nakon poziva funkcije potrebno je ispisati dobivenu matricu.

Funkcija treba imati prototip:

```
int *stvari_matricu(int *ulaz, int n);
```

Primjer rezultata rada funkcije za ulazno polje [1, 5, 3, 7]:

```
0 5 3 7
1 0 3 7
1 5 0 7
1 5 3 0
```

Rješenja:

1. zadatak

```
int PascalovTrokut(int redak, int stupac) {
    int clan;
    if (stupac == 0 || redak == stupac) {
        clan = 1;
    }
    //else if (stupac == 1 || (stupac + 1) == redak) {
    //    clan = redak;
    //}
    else {
        clan = PascalovTrokut(redak - 1, stupac - 1) +
            PascalovTrokut(redak - 1, stupac);
    }
    return clan;
}
```

2. zadatak

- a) $O(n^2)$, $\Omega(n)$ (najbolji slučaj: silazno sortirano polje; najgori slučaj: uzlazno sortirano polje)
b) $O(2^n)$, $\Omega(2^n)$, $\Theta(2^n)$

3. zadatak

poziv f2	Povr. adresa f2	4
	a	4
	b	4
poziv f2	Povr. adresa f2	4
	a	4
	b	4
poziv f1	t	4
	d	4
	c	4
	Povr. adresa f1	4
	&A[0]	4
	&A[1]	4

4. zadatak

[illegible]

5. zadatak

```
int *stvari_matricu(int* ulaz, int n){
    int *rez;
    int i; int j;
    rez = (int*)malloc((n*n)*sizeof(int));
    for (i = 0; i < n; i++){
        for (j = 0; j < n; j++){
            if (i!=j)
                rez[i * n + j] = ulaz[j];
            else
                rez[i * n + j] = 0;
        }
    }
    return rez;
}

int main(){
    int ulaz[] = {1, 5, 3, 7};
    int *rez;
    int i; int j;
    rez = stvari_matricu(ulaz, 4);
    for (i = 0; i < 4; i++){
        for (j = 0; j < 4; j++){
            printf("%d ", rez[i * 4 + j]);
        }
        printf("\n");
    }
    free(rez);
    return 0;
}
```