

1. Šta je od sljedećih tvrdnji istinito.. Odgovor: na stog se pohranjuju argumenti funkcije (ostali ponuđeni odgovori su većinom bili vezani za tip i veličinu varijable)
2. Matrica 3x2 sa pokazivačem na prvo polje i u petlji se zbraja suma sa  $*(p + i*MAXSTUP)$ , dakle samo prve elemente redaka. Odgovor je bio 9.
3. U programu dinamički alociran niz od 10 elemenata koji treba povećati memoriju za još 5 elemenata, pa je to:  $(int *) \text{realloc}(\text{niz}, 15*\text{sizeof}(int))$ .
4. Shell sort i složenost: ovisi o sekvenci koju izaberemo.

$$f(n) \sim g(n) \text{ ako je } \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 1$$

- 5.
6. Analiza „a posteriori“ ovisi o svemu (i o brzini procesora, algoritmu bla bla).
7. 

```
for (i=0; i<n; i++, sum1+=i);
for (i=0; i<n; i++)
for (j=0; j<n; j++, sum2+=j);
```

```
for (i=0; i<n; i++)
for (j=0; j<n; j++)
for (k=0; k<n; k++, sum3+=k)
```

Apriorna složenost:  $n^3$

8. 

```
ispis(char *niz, int n) {
    if ( (*niz != '\0') && (n >= 0) ) {
        printf("%c", *niz);
        ispisd(niz+1, n-1); }
}
```

```
ispisd(char *niz, int n) {
    if ( (*(niz + n) != '\0') && (n >= 0) ) {
        printf("%c", *(niz+n));
        ispis1(niz, n-1); }
}
```

Što će se ispisati slijedećim pozivom funkcije?  
`ispisd("ABECEDA", 5);`

Odgovor: DAEBCE

\*Pripaziti kod poziva funkcije, nisi uvijek isti jer nekad stave da se prvo pozove druga funkcija pa je drugi rezultat.

9. Ako se koristi čitanje po blokovima, za koji podatak će najviše trebati vremena? Za zadnji.
10. Najbolji odabir stožera? Nisam sigurna kako je bilo ponuđeno, da li samo element na polovici ili baš je medijan spomenut, uglavnom su ostali odgovori prvi/zadnji element pa se da zaključiti.

U onom pitanju kad imate dvije funkcije `ispis1(int n)` i `ispis2(int n)`, ako vam se pojavi ELSE u funkciji `ispis1`, onda vam je točan odgovor da `ispis1` ispisuje 0, a `ispis2` ispisuje silazno sve brojeve od `n`. Znači **nemojte** ko ja odmah napamet zaokružiti odgovor da je prva funkcija uzlazno, a druga silazno, jer pitanje nije identično onom u prošlim blicovima.

Ostala pitanja se većinom ponavljaju... a na ovom treba biti oprezan - pitanje asimptotskog **prosječnog** vremena:

```
funkcija(int x){..  
int f;  
for (i=0; i<n^2; i++){  
if (i==x) break;  
f+=i;  
} }
```

Rješenje je  $\sim n^2/2$  jer se traži prosječno, jer za prosjek gledamo da je x negdje na pola puta do  $n^2$ .

Imao sam jedan sa quick sortom. znaci ovo je polje: 4 15 1 9 8 13 17 20 i i treba odrediti kako polje izgleda nakon 1. rekurzivnog poziva algoritma.

točno je: 1 4 15 9 8 13 17 20 (nemam pojma kako)

i jos jedan sa Shell sortom lagani, ostalih 8 na istu foru kao i proslah godina

kod mene je bilo složenost fibonaccija (6) i onako nacrtano.. ugl, odg je bio  $2^n$ .

kod za funkciju:

$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$

$T_0 = 1$

$T_1(x) = x$

odgovori se razlikuju u zadnjem redu, točno je:  $\text{return } 2 * x \text{ Cheb}(x, n-1) - \text{Cheb}(x, n-2)$

ono u datotekama, pogledajte si a, a+, w, r+,r.. odgovor je a.

-složenost merge sorta ( $n \log n$ )

-složenost quick sorta za najbolji slučaj ( $n \log n$ )

-asimptotska složenost, neki primjer sličan onima na materijalima

-koja je netocna tvrdnja za naredbu `char *fgets` (ne sjećam se sta je bilo ponudeno)

-neki zadatak sa rasponom vrijednosti u pretincu adrese X ako su adrese pretinaca iz nekog intervala (vec videno na materijalima)

Od mojih novih je bila caka s propadanjem u rekurziju pa je trebalo jedino primjetiti da se prvo ispisuje ona "najdublja" rekurzija pošto je prvo na stogu. Onda je bio zadatak da je prvi član niza stožer i kako će izgledati nakon prve rekurzije u quicksortu.

Par mojih pitanja:

- 1) Apriori složenost algoritma za binarno pretraživanje?  $\log n$
- 2) program gdje vanjska for petlja ide do  $n$ , a unutrašnja do  $n^3$ ; pita se apriori složenost. Rj:  $n^4$
- 3) program čiji je isječak:

```
n=1;
while (fgets(s,512,f))
n++;
datoteka iz koje se čitalo je: aa\nb\nc\nddddddd\n
pita se koliki je n na kraju. rješenje 5
```

- 1) Asimptotska složenost za:

```
for (i=0;i<1;i++) for (j=0;j<1;j++)<naredbe...>
```

( $O(1)$ )

- 2) Apriori u najboljem slučaju za:

```
int k=nešto;
for(i=0;i<n;i++){
if (i*n==k)
for (j=0;j<m;j++)
k*=j;
break;}

```

( $\sim m$ )

- 3) Quicksort, kako izgleda niz nakon određivanja pivota po medijanu trojke i sakrivanja pivota
- 4) Insertion sort, mergesort, bubblesort, napredniji bubble sort, Shell sort - koji je nestabilan? (Shell)
- 5) Što će se ispisati?

```
char c;
FILE *fp=fopen("datoteka.txt", "r")
while((c=fgetc(fp))>0)putchar(c);
```

U datoteci:

l\0spitni primjer\n tekstualne datoteke