

1. Izračunajte vremensku složenost rekurzivne funkcije

```
unsigned faktorijel(unsigned n)
{ if (n == 0) return 1;
  else return n*faktorijel(n-1);
}
```

Rešenje:

Neka $T(n)$ označava broj instrukcija koje zahteva poziv funkcije faktorijel za ulaznu vrednost n .

Za $n = 0$, važi $T(n) = 2$ (jedno poređenje i jedna naredba return).

Za $n > 0$, važi $T(n) = 4 + T(n-1)$ (jedno poređenje, jedno množenje, jedno oduzimanje, broj instrukcija koje zahteva funkcija faktorijel za argument $n-1$ i jedna naredba return).

Nehomogena jednačina $T(n) = 4 + T(n - 1)$ se može rešiti svođenjem na homogenu jednačinu.

Iz $T(n) = 4 + T(n - 1)$ i $T(n + 1) = 4 + T(n)$ sledi $T(n + 1) - T(n) = T(n) - T(n - 1)$ i $T(n + 1) = 2 T(n) - T(n-1)$.

Karakteristična jednačina $t^2 = 2t - 1$ ima dvostruko rešenje $t_1 = 1$, pa je opšte rešenje oblika

$T(n) = a \cdot 1^n + b \cdot n \cdot 1^n = a + nb = O(n)$.

2. Dokazati korektnost prethodnog fragmente programa (pomoću principa matematičke indukcije).

3. Koja je vremenska složenost narednog koda

```
for(i=0; i<n; i++)
  for(j=0; j<i; j++)
    { if(j==n) break; s++; }
```

4. Niz ima k elemenata sa vrednostima u rasponu $[-n, n]$ i nije sortiran. Složenost najefikasnijeg postupka pretrage za takav niz je $\Theta(???)$.