**Ime:**

**Prezime :**

1. Za idući kôd odrediti vremensku složenost T(N) i vremensku klasu složenosti O(f(N)) (10 bodova):

int f(int \*niz, int n) {

int i = 0, sum = 0;

while(i < n) {

int j = 1;

while (j < n) {

sum += niz[j];

j = 3\*j;

}

i++;

}

return sum;

}

1. Procijeniti složenost O() algoritma koji računa operaciju za niz brojeva dužine N. Svi brojevi su cijeli brojevi u rasponu [1, K]. Algoritam traži koji se broj pojavljuje najviše puta u nizu. Algoritam se može napisati na tri načina:
2. Za svaki broj u nizu, broji se njegovo pojavljivanje u cijelom nizu i pamti se broj sa najviše pojavljivanja.
3. Sortiramo niz i brojimo koliko puta se pojavljuje svaki broj bez vraćanja.
4. Za svaki broj od 1 do K, broji se njegovo pojavljivanje u cijelom nizu i pamti se broj sa najviše pojavljivanja.

Složenost za svaki način izraziti u N i K. Možete pretpostaviti da se za sortiranje koristi quicksort.

(20 bodova)

1. Napisati funkciju koja prima jednostruko vezanu listu (svaki element sadrži broj). Funkcija će vratiti pokazivač na element sa najvećim brojem. (25 bodova)
2. Napisati funkciju koja prima vezanu listu sa barem tri elementa. Funkcija će izbrisati zadnja tri elementa iz liste. (35 bodova)
3. Prikazati quicksort algoritam na nizu { 6, 9, 6, 2, 2, 11, 3, 15 }. (10 bodova)