**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/2) {

sum += niz[i];

int j = 0;

while (j < n/2) {

sum++;

j++;

}

i++;

}

return sum;

}

1. Procijeniti klasu složenosti O() algoritma koji računa histogram za niz brojeva dužine N. Histogram je vezana lista gdje svaki element sadrži jedan broj iz niza i koliko puta se taj broj pojavljuje u nizu. Ukupno imamo M različitih brojeva u nizu tako da će histogram na kraju imati M elemenata. Algoritam se može napisati na tri načina:
2. Prolazimo kroz niz i za svaki broj provjerimo da li se već nalazi u listi. Ako da, uvećamo brojač za taj broj. Ako ne, dodamo ga na početak liste.
3. Sortiramo niz. Nakon toga prolazimo kroz sortirani niz i za svaki novi broj dodamo element na početak liste i brojimo koliko puta se pojavljuje dok ne naiđemo na idući broj.
4. Prolazimo kroz niz i za svaki broj provjerimo da li se već nalazi u listi. Ako da, uvećamo brojač za taj broj. Ako ne, dodamo ga po redu (veličini broja) u listu.

(20 bodova)

1. Napisati funkciju koja prima jednostruko vezanu listu (svaki element sadrži broj). Funkcija će ispred svakog elementa dodati novi element sa brojem 0. Kolika je složenost funkcije O()? (35 bodova)
2. Napisati funkciju koja prima vezanu listu (svaki element sadrži broj) i provjerava da li su svi elementi u listi jedinstveni. Kolika je složenost funkcije O()? (25 bodova)
3. Prikazati quicksort algoritam na nizu { 5, 1, 9, 2, 7, 6, 3, 4, 8 }. (10 bodova)