**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 = 1, sum = 0;

while(i < n) {

int j = 0;

while (j < n) {

sum += niz[j];

j++;

}

i \*= 2;

}

return sum;

}

1. Procijeniti složenost O() algoritma koji računa operaciju za 3 skupa (3 niza brojeva) dužine N, M i K. Algoritam kreira novi skup tako da uzima sve element iz prvog skupa (dužine N) koji se pojavljuju u drugom skupu (dužine M), ali se ne pojavljuju u trećem skupu (dužine K). Algoritam se može napisati na dva načina:
2. Nijedan niz nije sortiran. Prolazimo kroz prvi niz i za svaki broj provjerimo da se nalazi u drugom nizu i da se ne nalazi u trećem nizu.
3. Sva tri niza su sortirana. Za provjere koristimo binarnu pretragu.

Složenost izraziti u N, M i K.

(20 bodova)

1. Napisati funkciju koja prima jednostruko vezanu listu (svaki element sadrži broj). Funkcija će obrisati svaki element čiji je broj veći od broja u prethodnom elementu. Kolika je složenost funkcije O()? (35 bodova)
2. Napisati funkciju koja prima vezanu listu (svaki element sadrži broj) i jedan broja. Funkcija vraća pokazivač na element gdje se nalazi taj broj ili NULL ako se broj ne nalazi u listi. Kolika je složenost funkcije O()? (25 bodova)
3. Prikazati quicksort algoritam na nizu { 6, 2, 10, 3, 8, 7, 4, 5, 9 }. (10 bodova)