**Ime:**

**Prezime :**

**Matični broj:**

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 < 2 \* n) {

int j = 0;

while (j < n / 2) {

sum += niz[j];

j++;

}

i++;

}

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 su veći od svih elemenata u drugom skupu (dužine M), ali i manji od svih elemenata 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 li postoji veći broj u drugom ili manji broj u trećem nizu.
3. Pronađemo najveći i najmanji broj u drugom i trećem nizu. Nakon toga iz prvog niza uzmemo sve brojeve veće od najvećeg iz drugog niza i manje od najmanjeg iz trećeg niza.
4. Sortiramo silazno drugi niz i uzlazno treći niz. Nakon toga iz prvog niza uzmemo sve brojeve veće od prvog broja drugog niza i manje od prvog broja trećeg niza.

Složenost izraziti u N, M i K. Možete pretpostaviti da će pretrage u (a) proći pola niza, a da se za sortiranje u (b) koristi quicksort.

(20 bodova)

1. Napisati funkciju koja prima jednostruko vezanu listu (svaki element sadrži broj). Funkcija će između svaka dva parna susjedna broja 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 jedan dodatni broj. Funkcija vraća pokazivač na zadnji element u listi koji je veći od dodatnog broja ili NULL ako se takav broj ne nalazi u listi. Kolika je složenost funkcije O()? (25 bodova)
3. Prikazati quicksort algoritam na nizu { 7, 3, 11, 2, 9, 6, 2, 5, 8 }. (10 bodova)