**Ime Prezime:**

1. Za idući kôd odrediti T(N) i O(N) (15 bodova):

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

int i = n, sum = 0;

while(i > 0) {

int j = 0;

while(j+i < n) {

sum += niz[j+i];

j++;

}

i--;

}

return sum;

}

1. Procijeniti klasu složenosti O() algoritma koji u nizu dužine N traži dva broja čiji zbroj daje 0. Algoritam samo vraća True ili False ovisno o tome da li postoje takva dva broja u nizu. Algoritam se može napisati na dva načina:
2. Niz nije sortiran. Svaki broj u nizu pokušamo zbrojiti sa svakim od preostalih brojeva u nizu.
3. Niz je sortiran. Postavimo dva indeksa na početak i kraj niza (L i R). Ako je zbroj brojeva na ta dva indeksa veći od 0, umanjujemo R. Ako je zbroj manji od nula, uvećavamo L. Algoritam se prekida ako je broj na indeksu L pozitivan, ako je broj na indeksu R negativan ili ako L i R postanu jednaki.

(15 bodova)

1. Napisati funkciju koja prima jednostruko vezanu listu (svaki element sadrži broj). Funkcija provjerava da li je lista sortirana uzlazno. Funkcija vraća 0 ako nije sortirana ili 1 ako je. Kolika je složenost funkcije O()? (25 bodova)
2. Napisati funkciju koja prima listu i dva indeksa. Funkcija iz liste briše sve elemente koji nisu između ta dva indeksa (isključujući same indekse). Funkcija vraća izmijenjenu listu. Može se pretpostaviti da su indeksi legalni (unutar dužine liste). Kolika je složenost funkcije O()? (35 bodova)
3. Prikazati quicksort algoritam na nizu { 9, 6, 7, 2, 1, 8, 3 }. (10 bodova)