**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 <= n) {

int j = 1;

while (j < 2\*n) {

sum += niz[j];

j = j + j;

}

i++;

}

return sum;

}

1. Procijeniti složenost O() algoritma koji računa operaciju za niz brojeva dužine N. Algoritam provjerava da se nijedan broj u nizu ne ponavlja. Algoritam se može napisati na tri načina:
2. Prolazimo kroz prvi niz i za svaki broj provjeravamo da se ne pojavljuje nigdje drugo u nizu.
3. Prolazimo kroz prvi niz i za svaki broj provjeravamo da se ne pojavljuje nigdje drugo u ostatku niza.
4. Sortiramo niz uzlazno. Zatim za svaki broj niza provjerimo da je različit od idućeg broja u nizu.

Složenost za svaki način izraziti u N. Možete pretpostaviti da se za sortiranje koristi mergesort.

(20 bodova)

1. Napisati funkciju koja prima jednostruko vezanu listu (svaki element sadrži broj). Funkcija će provjeriti da li je lista sortirana uzlazno. (25 bodova)
2. Napisati funkciju koja prima vezanu listu (svaki element sadrži broj) i jedan dodatni broj. Funkcija će dodati novi element na predzadnje mjesto ako lista nije prazna, ili na prvo mjesto ako je lista prazna. (35 bodova)
3. Prikazati quicksort algoritam na nizu { 7, 3, 15, 2, 11, 6, 2, 6, 9 }. (10 bodova)