**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 \* n) {

sum += niz[j];

i++;

}

int j = 1;

while (j < 2 \* n) {

sum += niz[j];

j++;

}

return sum;

}

1. Procijeniti složenost O() algoritma koji iz neke datoteke čita elemente (brojeve) u nekakav niz. Za svaki pročitani element algoritam provjerava da li se element već nalazi u nizu. Broj pročitanih elemenata je N. Algoritam se može napisati na dva načina:
2. Niz nije sortiran i pri svakom čitanju se provjeravaju svi prethodno pročitani elementi.
3. Niz se drži sortiran i pri svakom čitanju se novi element dodaje na odgovarajuće mjesto (pa je to mjesto potrebno osloboditi pomicanjem elemenata). Pri dodavanju se ujedno i provjeri da li se element pojavljuje u nizu.

Složenost izraziti u N.

(20 bodova)

1. Napisati funkciju koja prima jednostruko vezanu listu (svaki element sadrži broj) i provjerava da li su svi elementi u listi različiti. Funkcija vraća 1 ako jesu ili 0 ako nisu. (25 bodova)
2. Napisati funkciju koja prima vezanu listu (svaki element sadrži broj) i jedan dodatni broj. Funkcija će dodati novi element sa tim brojem nakon najvećeg broja u listi. (35 bodova)
3. Prikazati quicksort algoritam na nizu { 7, 5, 11, 4, 10, 7, 6, 3, 9 }. (10 bodova)