**Ime Prezime: Grupa A**

1. Opisati tri algoritma koji provjeravaju da li su dva stringa anagrami („abcab“ je anagram od „bbcaa“). Stringovi su sastavljeni samo od malih slova. Prvi algoritam ne sortira stringove, drugi algoritam sortira jedan od dva stringa, treći algoritam sortira oba stringa (sortiraju se sa qsort). Za svaki algoritam odrediti vremensku složenost O u ovisnosti o dužini stringova N i na temelju O složenosti pojedinih operacija (npr. provjeri da li je slovo u stringu linearnom pretragom).
2. Za iduće funkcije odrediti T(n) i O(f(n)), gdje je n dužina niza:

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

int z = 0;

for (int i = n; i > 0; i--) {

for(int j=n; j > 0; j=j/2) {

z += niz[j];

}

}

return z;

}

void ispis(int \*niz, int n) {

for (int i = 0; i < n; i++) {

for (int j = i; j < n; j++)

printf("%d ", niz [j]);

for (int j = 0; j < i; j++)

printf("%d ", niz [j]);

}

}

1. Prikazati mergesort i quicksort na idućem nizu: { 3, 6, 9, 6, 2, 1, 2, 8, 8 }.

Za iduće zadatke sa listama je definirana iduća struktura:

typedef struct Element {

int broj;

struct Element \*next;

} Element;

**U svakom zadatku naznačiti da li se koristi „dummy“ head element.**

1. Napisati funkciju koja prima listu i „vraća“ dvije liste preko parametara. Dvije nove liste nastaju od prve liste tako da u jednu idu svi neparni brojevi, a u drugu svi parni brojevi. Npr. za listu {1, 3, 6, 3, 2, 4}, funkcija vraća liste {1, 3, 3} i { 6, 2, 4 } (redoslijed u novim listama je nebitan).

1. Napisati funkciju koja prima listu i vraća (izmijenjenu) listu. Funkcija iz liste briše zadnji element kojemu je broj manji od 0.