

Izvještaj za Drugi domaći zadatak  
Strukture podataka i algoritmi 2024/2025, Matijaš Jelena (1102/23)

Simulacija Taksi prevoza:

- Grad se sastoji od adresa koje predstavljaju čvorove u težinskom grafu. Graf je predstavljen matricom susjednosti *streets* govori koliko vremena je potrebno da se dođe od jednog do drugog čvora, pri čemu se smatra da čvorovi  $u$  i  $v$  nisu direktno povezani kada je  $streets[u][v]$  jednako nuli. Grad je predstavljen klasom *City* čiji su atributi matrica susjednosti *streets* i broj čvorova u grafu *numAddresses*. Matrica susjednosti se pri pokretanju programa učitava iz fajla.
- Podaci o svim taksijima čuvaju su u čvorovima jednostruko ulančane liste tipa *LIST*. Svaki čvor date liste ima informacije u identifikacionom stringu taksija, broju obavljenih vožnji, vremenu koje je taksi obavljajući vožnje, trenutnu adresu na kojoj se nalazi i pokazivač na idući element u listi. Na početku se svi taksiji iz ulaznog fajla učitavaju u listu te se njihovi podaci o broju obavljenih vožnji, trenutnoj adresi i ukupnom vremenu vožnji ažuriraju ukoliko taksi obavi narednu vožnju.
- Struktura *TAXI* ima sve podatke kao što ima struktura *LIST* osim što umjesto atributa *traveled* sadrži atribut *distance* koji predstavlja udaljenost taksija od adrese na koju treba otići. Takođe, nema samo pokazivač *next* već ima dva pokazivača na svoje lijevo – *left* i desno – *right* podstablo. Klasa *Tree* sadrži atribut tipa strukture *TAXI – root*.
- Upiti korisnika se učitavaju u dva niza *from* i *to* koji sadrže adresu na kojoj se nalazi korisnik i adresu na koju korisnik želi da ode, redom. Upiti se obavljaju jedan po jedan pri čemu se prvo izračunavaju udaljenosti svih taksija od trenutne adrese korisnika pomoću Dijkstra algoritma implementiranog u funkciji *shortestPath*. Ukoliko postoji put od taksija do adresu korisnika, ukoliko se taksi ne nalazi na samoj adresi korisnika i ukoliko je dostupan dodaje se u binarno stablo klase *Tree*. Nakon što se odrede sve udaljenosti taksija od adresu korisnika traži se najbliži taksi u binarnom stablu i šalje se na adresu korinika pri čemu se ispisuje putanja kojom se kreće. Zatim, određuje se najbliži put od trenutne adresu korisnika do adresu na koju želi otići i onda se korisnik prevozi do destinacije. Kada završi vožnju, proputovano vrijeme taksija se povećava, kao i broj obavljenih vožnji, a dostupnost se ažurira u odnosu na određen uslov.
- Nakon svih obavljenih vožnji na kraju simulacije će se ispisati ukupno vrijeme čekanja korisnika na taksi, ukupno vrijeme prevoza za svakog korisnika, broj obavljenih vožnji po taksiju i ukupan pređeni put svih vozila.

Rezultati izvršavanja nalaze se u *Output.txt* fajlu.