

STRUKTURE PODATAKA I ALGORITMI

DRUGI DOMAĆI ZADATAK

(datum objave: 06. 12. 2024.)

Grad je predstavljen kao graf, gdje su čvorovi adrese, a grane predstavljaju ulice između tih adresa. Svako taksi vozilo je na određenoj adresi i treba da bude dodijeljeno korisnicima koji traže prevoz. Svaki korisnik šalje upit sa adresom odakle treba biti prevezen i adresom na koju treba ići. Cilj je dodijeliti najbrži taksi na osnovu njegovog trenutnog položaja, koristeći najkraći put do korisnika.

Upiti korisnika dolaze u realnom vremenu i treba ih brzo obraditi. Upiti korisnika i raspoređivanje taksi vozila se optimizuju korišćenjem grafova za izračunavanje najkraćih puteva i binarnih stabala za brzo upravljanje slobodnim taksi vozilima.

Specifikacije:

1. Graf grada

- Grad je predstavljen kao neusmjereni težinski graf sa n čvorova, gdje n predstavlja broj adrese u gradu.
- Svaka adresa (čvor) povezana je s drugim adresama putem ulica (grana) sa određenim težinama, koje predstavljaju vrijeme potrebno za putovanja između adresa izraženo u minutama.
- Matrica susjednosti sadrži informaciju o vremenu putovanja između adresa. Ako je vrijednost 0, to znači da ne postoji direktna veza između tih adresa.

2. Taksi vozila:

- Grad ima t taksi vozila. Svako vozilo se nalazi na određenoj adresi i ima vrijeme dolaska (koliko vremena treba da stigne do korisnika – ovo vrijeme se proračunava na osnovu trenutne adrese i adrese na kojoj se korisnik nalazi), trenutnu adresu i jedinstveni identifikator.
- Taksi vozila se raspoređuju korisnicima na osnovu najbržeg vremena dolaska (najbliži taksi do korisnika).
- Svako taksi vozilo može biti ili *slobodno* ili *zauzeto*. Ako je vozilo zauzeto, mora čekati dok završi prevoz korisnika prije nego što postane ponovo slobodno.

3. Upiti korisnika:

- Svaki korisnik šalje upit u obliku: Trenutna adresa korisnika + Adresa odredišta (gdje želi da ide).
- Upiti korisnika dolaze u realnom vremenu, pa je potrebno brzo dodijeliti najbliže slobodno vozilo. Ako su svi taksiji zauzeti, korisnik mora čekati da se neko vozilo oslobodi.

4. Algoritmi:

- Dijkstra algoritam - Koristi se za izračunavanje najkraćeg puta između adresa (korisnikove trenutne lokacije i adrese na koju treba da ide).
- Binarno stablo pretrage (BST) - Koristi se za brzo dodjeljivanje slobodnog taksija korisnicima prema vremenu dolaska, tako da se najbrži taksi (onaj koji može da stigne

prvi) dodjeljuje korisniku. Za svaku vožnju kreira se novo BST stablo i vozilo se dodjeljuje korisniku kao BST-MIN – vozilo koje je najbliže korisniku, a koje je slobodno. Ukoliko BST-MIN nije slobodan, traži se njegov sljedbenik, i td.

Ulazni podaci su:

- CSV fajl sa matricom susjednosti - Sadrži $n \times n$ matricu vremena putovanja između adresa.
- CSV fajl sa upitima korisnika - Svaki red sadrži trenutnu adresu korisnika i adresu na koju treba biti prevezen.
- Tekstualni fajl sa taksi vozilima, u kom se svaki red sastoji od jedinstvenog identifikatora vozila i njegove trenutne adrese.

Simulacija treba da se odvija na sljedeći način:

- Učitavaju se ulazni podaci i smještaju u odgovarajuće strukture.
- Prolazi se kroz upite korisnika redom i za svaki upit korisnika, potrebno je pretražiti binarno stablo taksi vozila i dodijeliti najbrži slobodan taksi. Nakon dodjele, taksi se stavlja u stanje zauzet.
- Izračunati najkraći put od trenutne lokacije taksija do korisnika i zatim do odredišta koristeći Dijkstra algoritam.
- Na konzoli treba prikazati dodijeljena taksi vozila korisnicima i vrijeme dolaska taksija za svaki upit korisnika.
- Za vrijeme trajanja vožnje na konzolu se prikazuje putanja kojom je vozilo odvezlo korisnika u obliku čvor1-čvor12-....
- Kada taksi završi prevoz, taksi vozilo se ponovo označava kao slobodno i informacija o završenoj vožnji se prikazuje na konzoli.
- Na kraju simulacije, potrebno je prikazati:
 - Ukupno vrijeme čekanja korisnika na taksi.
 - Ukupno vrijeme prevoza za svakog korisnika.
 - Broj obavljenih vožnji po vozilu.
 - Ukupan pređeni put svih vozila.

Napomene: Studenti su, pored rješenja domaćeg zadatka u vidu izvornog koda, lično kreiranih CSV datoteka, tekstualne datoteke i izvršnog fajla, obavezni da predaju izveštaj sa logovima na dva primjera izvršavanja (jedan primjer da koristi ulazne datoteke sa *Moodle*-a, a drugi sopstvene datoteke), kao i kratak opis urađenog (do jedne stranice). Logovi mogu biti priloženi kao *screenshot* ekrana nakon izvršavanja programa, ali tako da obuhvate sve faze simulacije ili kao *output* dokument u tekstualnom formatu, koji će predstavljati kopiju prikazanog na konzoli u toku simulacije. Domaći zadatak se predaje kao jedna arhiva imenovana po principu **Ime-Prezime-BrojIndeksa** na *Moodle* link za predaju, koji će biti aktivan od sljedeće sedmice. **U skladu sa već opisanim u propozicijama predmeta, zadatke je potrebno raditi samostalno i zadaci će biti podvrgnuti detekciji sličnosti.**

Rok za predaju domaćeg zadatka: 16. 12. 2024. godine do 16:00.