

STRUKTURE PODATAKA I ALGORITMI

DRUGI DOMAĆI ZADATAK

(datum objave: 03. 12. 2025.)

Napisati program u C ili C++ koji simulira sistem prikupljanja podataka iz mreže senzora u pametnom gradu. Grad je modelovan kao **neusmjeren, težinski graf**: čvor = senzor, grana = komunikaciona veza, težina = vrijeme/trošak prenosa.

Cilj je: (1) konstruisati **minimalno razapinjuće stablo (MST)**, (2) stablo **ukorijeniti** u čvoru centrale, (3) u jednom prolazu izračunati **opterećenja grana** ka korijenu, te (4) u **izvornom grafu** pronaći **rezervne rute** (najkraće puteve) pri kvaru odabrane grane stabla.

Program treba da uradi sljedeće:

1. Učitavanje mreže senzora (grafa)

- Ulaz u program predstavlja **matrica susjednosti** grada/senzorske mreže dimenzija $n \times n$, upisana u tekstualni fajl.
- Matrica predstavlja **neusmjeren težinski graf**: vrijednost > 0 na poziciji (i, j) znači da između senzora i i j postoji direktna veza. Vrijednost predstavlja težinu.

2. Učitavanje liste senzora

- Iz posebnog **CSV fajla** učitati listu senzora. Svaki red CSV fajla predstavlja jedan senzor i sadrži sljedeće informacije:
 - identifikator senzora (cjelobrojna vrijednost koja odgovara indeksu u matrici),
 - tip senzora (npr. "temperature", "air_quality", "noise"),
 - kritičnost (cijeli broj u opsegu 0–100),
 - količina podataka koju senzor generiše u jedinici vremena (npr. KB/min).
- Radi lakšeg testiranja, identifikatore senzora u CSV fajlu generisati kao cjelobrojne vrijednosti koje odgovaraju indeksima čvorova u matrici susjednosti $(0, 1, 2, \dots, n-1)$.

3. Definisanje centralne stanice

- Centralna stanica (glavni server) se nalazi na unaprijed zadatom čvoru grafa (npr. 0), ili se indeks čvora centrale unosi sa tastature.
- Svi senzori u mreži u konačnoj hijerarhiji treba da šalju podatke ka ovoj centralnoj stanici.

4. Izgradnja minimalne mreže (stablo iz grafa)

- Pošto realna mreža senzora može imati više veza nego što je neophodno, potrebno je iz postojeće mreže izgraditi **minimalno razapinjuće stablo (MST)** nad svim čvorovima grafa. Za izgradnju MST-a koristiti jedan od klasičnih algoritama: **Primov** ili **Kruskalov** algoritam.
- Dobijeno stablo predstavlja **najjeftiniju infrastrukturu** koju grad mora održavati kako bi svi senzori bili povezani.

5. Korijenjenje stabla u centralnoj stanici

- Ukorjenjivanje pretvara MST u usmjereno hijerarhijsko stablo sa smjerom ka korijenu (centralnoj stanici). Jednim BFS/DFS obilaskom od *root* postaviti $parent[child] = u$ pri prvoj posjeti susjedu *child* iz *u*. Time svaka ivica dobija orijentaciju dijete \rightarrow roditelj.

- Nakon korjenjenja, svaka veza u stablu treba da ima jasno definisan smjer: od senzora ka centrali (od djeteta ka roditelju).
- Stablo je potrebno ispisati u hijerarhijskom obliku (npr. uvučeni prikaz: korijen → djeca).

6. Računanje opterećenja grana u stablu

- Pošto svi senzori šalju svoje podatke prema centrali baš preko tog stabla, svaka grana u stablu treba da prenese zbir podataka svih senzora koji se nalaze “ispod” te grane.
- U **post-order DFS** obilasku (odozdo naviše) za svaku granu stabla izračunati:
 - ukupan obim podataka u njenom podstablu,
 - broj senzora u tom podstablu.
- Na konzoli prikazati one grane koje nose najveće opterećenje (moguće “usko grlo” sistema).

7. Pronalazak rezervnih ruta (u izvornom grafu)

- Za pronalaženje rezervnih ruta koristiti algoritam za pronalaženje najkraćeg puta (Dijkstrin algoritam ili Floyd–Warshall-ov algoritam).
- Rezervnu rutu potrebno je tražiti u slučaju **kvara grane**: ako ivica $x \rightarrow \text{parent}[x]$ otkáže, svi čvorovi u podstablu čvora x moraju naći **alternativni put do root** u originalnom grafu. Za svaki takav čvor v ispisati: najkraća ruta do root i njena dužina. Kvar grane simulirati tako što odaberete neku ivicu $x \rightarrow \text{parent}[x]$ iz MST-a i **privremeno je uklonite** iz originalnog grafa (postavite težinu na 0).

8. Izvještaj i upis u fajl

- Na kraju simulacije, na konzolu ispisati:
 - ukupnu cijenu/težinu minimalnog razapinjućeg stabla,
 - hijerarhijski prikaz stabla (centrala na vrhu),
 - opterećenje po granama,
 - rezultate pronalaska rezervnih ruta.

Zadatak treba da ilustruje razliku između rada sa grafom (stvarna mreža, najkraći put, povezanost), i rada sa stablom (logička/hijerarhijska mreža dobijena iz grafa, MST, korijenjeno stablo, sabiranje odozdo nagore).

Student treba da implementira bar jedan algoritam za najkraći put i jedan algoritam za izgradnju minimalnog stabla, kao i da pokaže kako se iz grafa izvodi stablo i kako se nad stablom rade dodatne analize (opterećenje, dubina, roditelj–dijete).

Primjer izgleda izlaznog fajla

```
MST ukupna tezina: 137
Korijen (centrala): 0

Hijerarhija (korijen 0):
0
  3
    5
    6
  1
    4
    7
```

2

Opterećenja grana (KB/min):

5 -> 3 : 18

6 -> 3 : 22

3 -> 0 : 40

4 -> 1 : 10

7 -> 1 : 12

1 -> 0 : 22

2 -> 0 : 15

Najopterećenija grana: 3 -> 0 (load=40 KB/min)

[SIMULACIJA KVARA] Grana 3 -> 0 je nedostupna.

REZERVNE RUTE (u originalnom grafu, Dijkstra od 0):

Cvor: 5

REZERVNA RUTA: 5 -> 3 -> 2 -> 0

Distanca: 8

Cvor: 6

REZERVNA RUTA: 6 -> 3 -> 1 -> 0

Distanca: 9

Cvor: 3

REZERVNA RUTA: 3 -> 1 -> 0

Distanca: 6

Napomene: Na Moodle stranici predmeta dostupni su testni fajlovi. Studenti su, pored rješenja domaćeg zadatka u vidu izvornog koda i lično kreiranih najmanje po dva tekstualna i CSV fajla, obavezni da predaju izveštaj sa logovima na primjeru izvršavanja, kao i kratak opis urađenog (do jedne stranice). Logovi mogu biti priloženi kao *screenshot* ekrana nakon izvršavanja programa, ali tako da obuhvati 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: 12. 12. 2025. godine do 16:00 časova.