

# Podatkovne strukture in algoritmi (2024/25)

## 6. domača naloga – programerski del

### Najcenejše vpeto drevo

Na strežniku za oddajanje nalog (<http://marmoset.famnit.upr.si>) je naložena šesta domača naloga, pri kateri je potrebno sprogramirati Primov algoritem za iskanje najcenejšega vpetega drevesa v uteženem enostavnem neusmerjenem grafu (enostaven graf nima zank in večkratnih povezav). V tej nalogi predpostavite, da je  $V = \{0, 1, \dots, n-1\}$  množica vozlišč grafa in da so uteži iz množice  $\mathbb{Z} \setminus \{0\}$ . Takšen utežen neusmerjen graf predstavimo z uteženo matriko sosednosti  $A \in \mathbb{Z}^{n \times n}$ , kjer je  $A$  simetrična matrika, to je,  $A[i][j] = A[j][i]$  za  $i, j \in V$ . Povezava med vozliščema  $i \in V$  in  $j \in V$  v grafu obstaja natanko tedaj, ko je  $A[i][j] \neq 0$ ; če v grafu obstaja povezava med  $i$  in  $j$ , potem je njena utež enaka  $A[i][j]$ . Sledite spodnjim navodilom:

V razredu Prim parameter `int[] [] data` določa uteženo matriko sosednosti, kot je opisana zgoraj, parameter `int n` pa njeno velikost. V razredu Prim najprej implementirajte konstruktor `Prim(int[] [] d)`. Nato v tem razredu implementirajte naslednje metode:

- (i) `addEdge(int i, int j, int d)`, ki v graf doda povezavo z utežjo  $d$  med vozlišči  $i$  in  $j$ .
- (ii) `MSTcost()`, ki vrne ceno minimalnega vpetega drevesa.
- (iii) `prim(int s)`, ki vrne minimalno vpeto drevo s korenom v vozlišču  $s$ . Minimalno vpeto drevo je podano s poljem staršev  $P$  – starš vozlišča  $i \in V$  v drevesu je enak  $P[i]$ . Predpostavite, da je starš korena  $s$  vedno enak 0, to je,  $P[s] = 0$ .

Na določenem koraku lahko obstaja več povezav z minimalno utežjo (to se zgodi, recimo, če imajo vse povezave enake uteži). Naj bo  $T$  trenutno drevo na nekem koraku. Recimo, da obstaja  $r$  povezav z minimalno utežjo, ki vežejo  $T$  z (neobiskanimi) vozlišči, ki jih  $T$  ne pokriva. Potem izberete tisto minimalno povezavo, ki ima najmanjše neobiskano krajiščno vozlišče. Bolj natančno, naj bo  $e_i$  minimalna povezava, ki veže  $T$  z neobiskanim vozliščem  $v_i$ ,  $i = 1, 2, \dots, r$ , in naj velja  $v_1 < v_2 < \dots < v_r$ . Tedaj izberite povezavo  $e_1$ , to je,  $v_1$  postane obiskano vozlišče. Za iskanje vozlišča  $v_1$  (in posledično minimalne povezave  $e_1$ ) ni potrebno uporabiti minimalne kopice, dovolj učinkovito bo že, če naivno pregledate kar vsa vozlišča. V ta namen ni potrebno uporabiti prednostne vrste, lahko se preprosto sprehodite čez vsa vozlišča in med njimi poiščete ustreznega.

Rešitev naloge oddajte preko <http://marmoset.famnit.upr.si>. Vsi testi, tako javni kot skriti, so točkovno enakovredni. Rok za oddajo nalog je **nedelja, 19. januar 2025**.