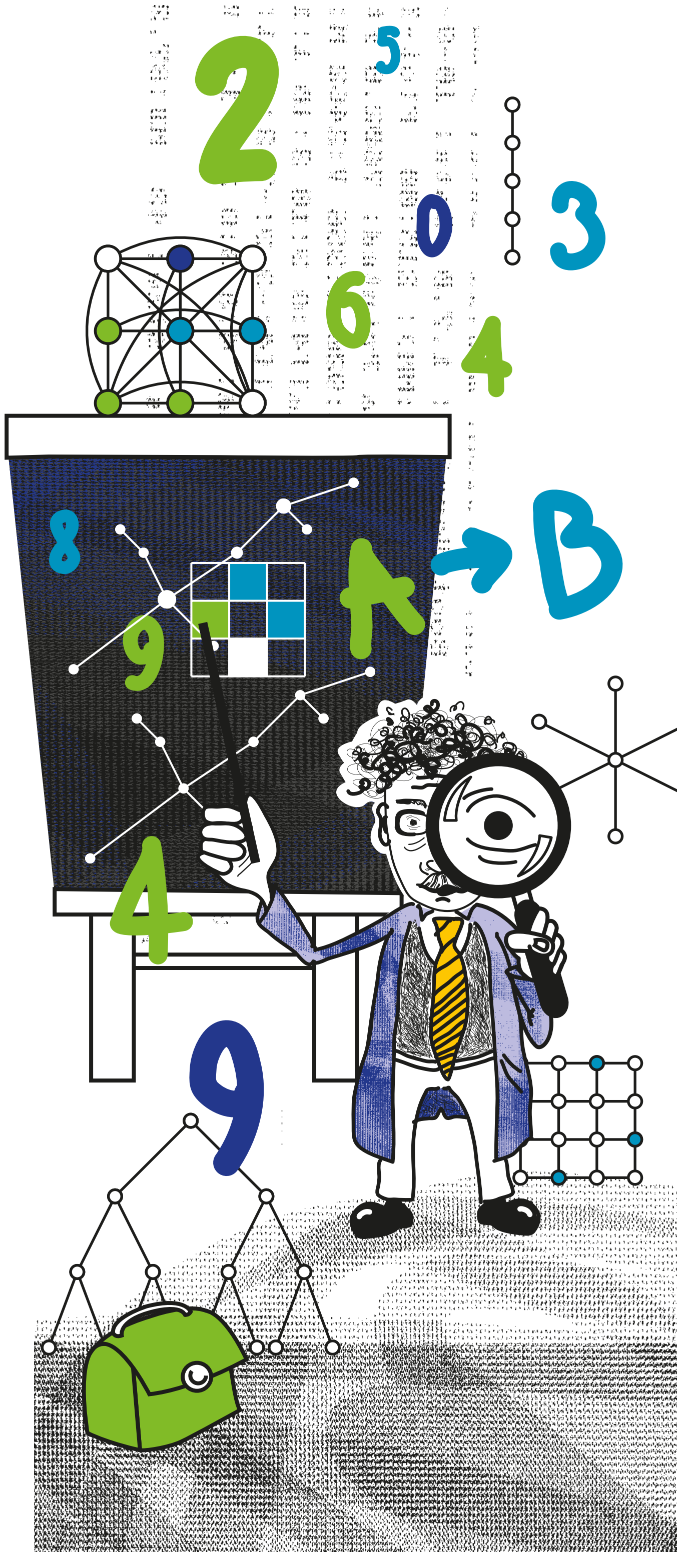
Dominantno število



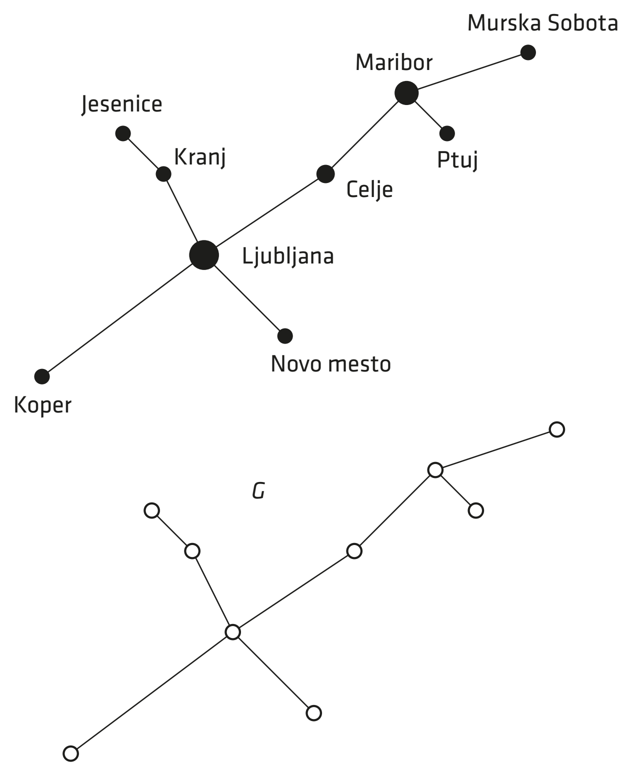
POLONA PAVLIČ

→ V času vse večjih podnebnih sprememb smo priča naraščajočemu številu vremenskih ujm. Zadnja takšna se je v Sloveniji zgodila novembra 2012, ko so naše kraje zajele katastrofalne poplave. V takih primerih je dobro, da imajo vsaj večja mesta zagotovljene organizirane oblike pomoči, kot je npr. Civilna zaščita. Zaradi vsesplošnega varčevanja je jasno, da je potrebno enote pomoči postaviti po državi tako, da jih bo potrebnih čim manj, hkrati pa bodo te preudarno postavljene. V nadaljevanju se bomo vprašali, kako najti finančno ter prostorsko čim bolj ugodno rešitev za poljubno razporeditev mest v katerih naj bi bile enote za pomoč, ter povezav med njimi. Za manjši primer bi z nekaj razmišljanja lahko hitro našli ugodno rešitev zastavljenega problema. Za večje primere pa ne moremo več z gotovostjo trditi, da je naša rešitev najboljša možna – morda obstaja razporeditev z manj enotami. Poskusimo problem zapisati formalno in ugotoviti, kako se reši za splošen primer.

Matematični pogled

V jeziku veje matematike, ki jo imenujemo *teorija grafov*, lahko probleme tega tipa zapišemo takole: mislimo si, da mesta predstavimo s točko, imenujemo jo *vozlišče*. Ceste, ki vodijo od enega do drugega mesta, pa predstavimo s črto, imenujemo jo *po-*

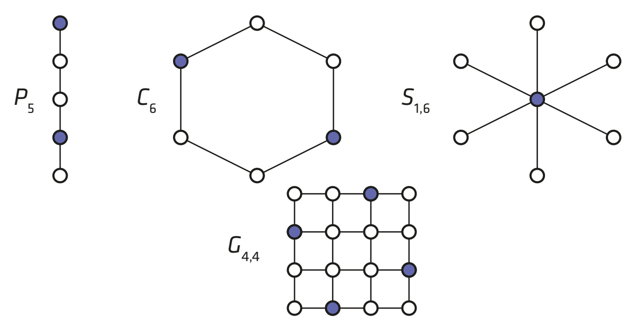
vezava. Množico vseh vozlišč V (angleško *vertices*), skupaj s pripadajočimi povezavami E (angleško *edges*), imenujemo *graf* in ga označimo z $G = (V, E)$. Če med dvema vozliščema obstaja povezava, pravimo, da sta *sosednji*. Primer grafa za predstavitev zemljevida slovenskih mest, ki jih neposredno povezuje avtocesta, je prikazan na sliki 1.



SLIKA 1.
Zemljevid večjih slovenskih mest in pripadajoči graf G

Za učinkovito postavitev organiziranih enot pomoči v mesta želimo pobarvati vozlišča pripadajočega grafa tako, da bo vsako vozlišče bodisi pobarvano samo bodisi bo sosednje s pobarvanim vozliščem. Pri tem pobarvano vozlišče pomeni, da je v tistem mestu na voljo enota pomoči, vsako mesto, ki ni pobarvano, pa mora imeti vsaj enega pobarvanega soseda. To pomeni, da lahko v to mesto dovolj hitro pride pomoč. Množici tako pobarvanih vozlišč grafa pravimo *dominantna množica*. Številu vozlišč v dominantni množici, ki ima med vsemi možnimi dominantnimi množicami najmanj elementov, pravimo *dominantno število* grafa in ga označimo $\gamma(G)$. Dominantno množico, ki ima najmanjše možno število vozlišč ($\gamma(G)$), imenujemo *najmanjša dominantna množica*. Za vozlišče grafa bomo včasih rekli, da je *dominirano*, če je bodisi v dominantni množici bodisi je sosednje z vozliščem iz dominantne množice grafa. Za graf slovenskih mest na sliki 1 je dominantno število 3. Poskusite sami poiskati optimalno razporeditev enot za ta primer. Preverite tudi trditev, da z dvema vozliščema ne moremo zadostiti pogoju.

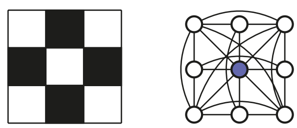
Na sliki 2 so obarvana vozlišča najmanjših dominantnih množic predstavljenih grafov. Prvi graf imenujemo pot na n vozliščih (v našem primeru je $n = 5$). Bi znali sami določiti dominantno število poti za poljubni n? Opazimo lahko, da mora biti v dominantni množici vsaj vsako tretje vozlišče. Tako pridemo do rezultata $\gamma(P_n) = \lceil \frac{n}{3} \rceil$. Podobno velja tudi za grafe, ki jih imenujemo cikli na n vozliščih, C_n. Na drugem primeru na sliki 2 je cikel C_6, katerega dominantno število je 2. Tretji graf na sliki je primer zvezde, $S_{1,n}$; v našem primeru je $n = 6$, v splošnem pa centralno vozlišče povežemo z n dodatnimi vozlišči. Očitno je za zvezdo vedno dovolj, da pobarvamo le centralno vozlišče. Torej je $\gamma(S_{1,n}) = 1$ za vsako naravno število n. Zadnji graf s slike 2 imenujemo mreža ali grid. Tukaj je določitev dominantega števila nekoliko bolj zahtevna. Za primer na sliki je rezultat 4, medtem ko je bil rezultat za mrežo dolžine n vozlišč in širine m vozlišč dolga leta neznan, dokazan pa leta 2011.



SLIKA 2.
Dominantna števila poti, cikla, zvezde in mreže

Preden nadaljujemo, premislite še naslednje: Ali lahko v vsakem izmed primerov na sliki 2 najdete tudi tako dominantno množico, ki jo sestavljajo druga vozlišča grafa tako da bo ta množica še vedno najmanjša?

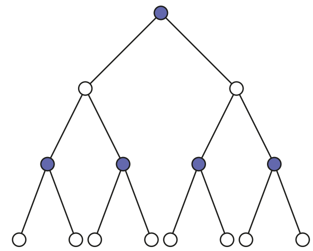
Dodajmo še, da je bil problem dominantnega števila pred časom že predstavljen v Preseku [1], vendar v nekoliko drugačni obliki. Avtor je predstavil, kako na šahovsko desko razporediti najmanjše število kraljic tako, da bodo kraljice pokrivale vsako polje šahovnice – bodisi je na polju kraljica bodisi lahko do polja pride z dovoljenim premikom (levo, desno, gor, dol ali po diagonalah). Če vsakemu polju šahovnice priredimo svoje vozlišče, dve vozlišči pa povežemo s povezavo, kadar lahko s kraljico pridemo iz enega na drugo polje, bo tako dobljeni graf ustrezal premikom kraljice po šahovski plošči. Dominanto število takega grafa pa bo ravno najmanjše število kraljic, ki jih moramo razporediti na polja tako, da bo vsako polje bodisi pokrito s kraljico bodisi pa bo kraljica do tega polja lahko prišla (glej sliko 3).



SLIKA 3.
Graf kraljice za šahovko desko velikost 3 × 3

Določitev dominantnega števila je težek problem

S temo dominacije grafov se ukvarja veliko število matematikov, napisanih je bilo že na tisoče člankov, zelo hitro pa je postalo jasno, da je problem določitve dominantnega števila grafa izjemno težek problem. Pravimo, da je ta problem, kot še mnogo zanimivih matematičnih problemov, NP-poln. To pomeni, da lahko za nekatere grafe z nekaj truda poiščemo dominantno število, vendar pa za poljubni graf tega ne moremo narediti, vsaj ne dovolj hitro (v polinomskem času). Dovolj hitro lahko samo preverimo, če je neka množica vozlišč res dominantna, to pa je tudi vse. Zato so se različni avtorji ukvarjali z določitvijo dominantnega števila nekaterih družin grafov. Tukaj bomo predstavili hiter algoritem (teče v linearnem času), ki poišče dominantno število drevesa. *Drevo* je povezan graf, ki je brez ciklov. Z drugimi besedami, v drevesu med poljubnima dvema vozliščema obstaja natanko ena pot (iz nekega vozlišča lahko prek ostalih vozlišč in povezav do drugega vozlišča pridemo na en sam način). Poseben primer drevesa, ki ga imenujemo tudi dvojiško drevo, je prikazan na sliki 4. Tudi graf slovenskih mest iz slike 4 je primer drevesa.



SLIKA 4.
Dvojiško drevo in njegova najmanjša dominantna množica

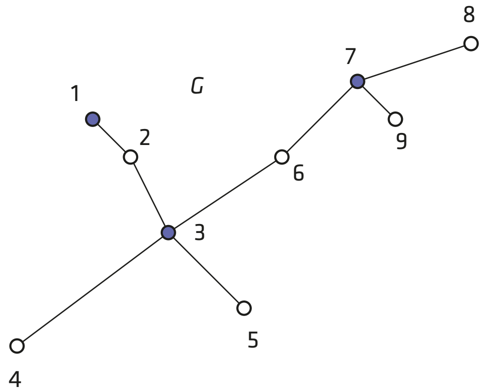
Algoritem za drevesa

Poglejmo sedaj kako najti najmanjšo dominantno množico za poljubno drevo T. Razdelimo njegova vozlišča v tri skupine, V_1, V_2 in V_3. Pri tem vsako izmed vozlišč spada v natanko eno izmed skupin. Množico V_1 imenujmo množica *prostih vozlišč*, vozlišča iz V_2 naj bodo *mejna*, vozlišča iz V_3 pa *potrebna*. Za množico vozlišč D pravimo, da je *opcijska dominantna množica*, če vsebuje vsa potrebna vozlišča (iz V_3) in dominira vsa mejna vozlišča (vsako vozlišče iz V_2 je sosednje z nekim iz D ali pa je v D). Za vozlišča iz V_1 ni pogojev, torej v opcijski dominantni množici niso niti nujno dominirana, lahko pa so ali v D ali pa sosednja z nekom iz D.

Razmislite, da je v primeru, ko je $V_1 = \emptyset$, $V_2 = V$ in $V_3 = \emptyset$, opcijsko dominantno število grafa enako običajnemu dominantnemu številu grafa. Zato algoritem za iskanje velikosti najmanjše opcijske dominantne množice grafa zadošča za iskanje dominantnega števila grafa.

Sedaj imamo pripravljeno vse, da se lahko lotimo algoritma za iskanje dominantnega števila drevesa. Izberimo poljubno vozlišče drevesa, imenujmo ga koren, in ga oštevilčimo z 1. Nato vsa nadaljnja vozlišča po vrsti oštevilčimo z zaporednimi naravnimi števili – najprej vse njegove sosede, nato pa po vrstnem redu vse sosede sosedov, itd. Tak način oštevilčevanja vozlišč je bil pred kratkim predstavljen pod imenom *pregled v širino* tudi v Preseku [3]. Podrobneje si za graf slovenskih mest ta postopek lahko ogledamo na sliki 5. Nadalje tvorimo polje *Starš* takole: korenu priredimo vrednost 0 (Starš [1] = 0), nato pa za vsako vozlišče povemo, katero od že pregledanih vozlišč je njegov sosed (starš). Za drevo s slike 5 imamo:

i	1	2	3	4	5	6	7	8	9
Starš [i]	0	1	2	3	3	3	6	7	7



SLIKA 5.
Izvedba algoritma za drevo s slike 1

Sedaj bomo poskušali poiskati najmanjšo dominantno množico drevesa. To bomo naredili tako, da vsakemu vozlišču priredimo oznako, „prosto“, „mejno“ ali „potrebno“. To zapišemo v polje *Oznaka*. V začetku vsako vozlišče označimo kot „mejno“ in najmanjšo dominantno množico označimo z D ter jo zaenkrat pustimo prazno. Začnimo pri vozlišču n, torej zadnjem. Če je označeno kot „mejno“ (v začetku to gotovo je res), njegovega starša označimo kot „potrebno“, saj mora biti tudi to zadnje vozlišče dominirano. Nato nadaljujemo postopek na vozlišču $n-1$. Če je označeno kot „mejno“, njegovega starša označimo kot „potrebno“, če pa je oznaka vozlišča $n-1$ „potrebno“, to vozlišče dodamo v dominantno množico D ter njegovega starša označimo „prosto“, saj je to vozlišče že dominirano. Tak postopek nadaljujemo vse do korena. Če je koren označen „mejno“ ali „potrebno“, ga dodamo v dominantno množico, sicer pa smo že pred tem končali in dominirali celoten graf. Najmanjšo dominantno množico danega drevesa torej sestavljajo vsa vozlišča z oznako „potrebno“. Natančneje si lahko ta postopek ogledate v algoritmu 1.

Glede na to, da moramo v algoritmu samo prepotovati skozi vsa vozlišča drevesa (imamo le dve zaporedni zanki), bo ta algoritem zares tekel v zgoraj omenjenem linearnem času. Spremljajte, kako algoritem 1 teče na primeru s slike 5, in videli boste, kako pridemo do obarvanih vozlišč – ta predstavljajo potrebna vozlišča, torej tista, ki tvorijo dominantno množico. Razmislite, ali ta postopek zares deluje, z drugimi besedami razmislite o tem, da je rešitev vedno najmanjša dominantna množica danega drevesa. Tukaj bomo podrobnosti izpustili.

Algoritem 1 Dominantno število drevesa

Vhod: Drevo T, ki je predstavljeno s poljem Starš $[1 \dots n]$.
Izhod: Najmanjša dominantna množica D drevesa T, ki jo predstavljajo vozlišča i z Oznaka $[i]$ = "*potrebno*".

```
D ← ∅;
za i = 1 ... n naredi
    Oznaka [i] = "mejno";
konec
za i = n ... 2 naredi
    če Oznaka [i] = "mejno" potem
        Oznaka [Starš [i]] = "potrebno";
    sicer če Oznaka [i] = "potrebno" potem
        D ← D ∪ {i};
        če Oznaka [Starš [i]] = "mejno" potem
            Oznaka [Starš [i]] = "prosto";
        konec
    konec
konec
če Oznaka [1] = "mejno"ali Oznaka [1] = "potrebno" potem
    D ← D ∪ {1};
konec
```

Literatura

[1] B. Brešar, *Kraljice napadajo!* Presek, 27 (1) 1999, 16–21.

[2] T. W. Haynes, S. T. Hedetniemi in P. J. Slater, *Fundamentals of Domination in Graphs.* Marcel Dekker, New York, 1998.

[3] A. Taranenko, *Predstavitev grafa s seznamom sosedov in pregled v širino.* Presek, 39 (6) 2012, 27–30.

×××

www.presek.si