30. Trumpiausio kelio radimo uždavinys

Grafų teorija

Vytautas Traškevičius

VU MIF, 2016 m.

Uždavinio formuluotė

- Duotas **svorinis** grafas $G=(V,U,C)$, čia $C[1..m]$ – grafo briaunų (lankų) svorių masyvas.
- Rasti:
  1) trumpiausius kelius nuo viršūnės *s* iki visų likusių grafo viršūnių
  2) trumpiausią kelią nuo viršūnės *s* iki viršūnės *t*
- 2)-asis uždavinys yra 1)-ojo atskiras atvejis
- 1)-ojo uždavinio sprendimui naudojamas Dijkstros algoritmas
  - Jei grafe nėra neigiamo svorio briaunų

Dijkstros algoritmo idėja

Pradžioje:

- visos viršūnės nenudažytos;
- $d(s, s) = 0$, s – pradinė viršūnė;
- iki visų kitų keliai nerasti, kelių ilgiai lygūs begalybei;

Toliau:

- iš visų nenudažytų išsirenkame artimiausią s viršūnei viršūnę k;
- k nudažome;
- laikome, kad trumpiausias kelias iki k jau rastas;
- kiekvienai k nenudažytai kaimynei patikriname, ar patekti į ją iš k labiau apsimoka, nei eiti iki jos iki šiol rastu trumpiausiu keliu; jei taip, perskaičiuojame trumpiausio kelio iki jos ilgį;
- šį procesą kartojame, kol yra nenudažytų viršūnių.

Dijkstros algoritmas

Procedūra randanti trumpiausius kelius svoriniame grafe nuo viršūnės *s* iki likusių viršūnių

Parametrai:

- *s* – viršūnės, nuo kurios norime rasti trumpiausius kelius, numeris
- *n* – grafo viršūnių skaičius,
- *m* – grafo briaunų (lankų) skaičius (grafas gali būti ir orientuotasis)
- *c*[1..*m*] – grafo briaunų svorių masyvas

Procedūros rezultatai

Masyvas $d[1..n]$

- i-asis elementas $d[i]$ yra trumpiausio kelio nuo viršūnės s iki viršūnės i ilgis
- $d[s] = 0$

Masyvas $prec[1..n]$

- i-asis elementas $prec[i]$ nusako, iš kokios viršūnės kelias veda į viršūnę i
- $prec[i] = k$, jei kelias į viršūnę i veda iš viršūnės k
- $prec[s] = s$
- $prec[i] = 0$, jei i dar neaplankyta viršūnė

Procedūra

Pradžioje visos grafo viršūnės nenudažytos

sum = visų masyvo *c* elementų suma + 1 (*sum* laikoma begalybe)

Užnuliname *prec* masyvą

prec[*s*] := *s* (visi keliai prasideda viršūnėje *s*)

Visiems masyvo *d* elementams priskiriame *sum* (begalybę)

d[*s*] = 0

Kol yra nenudažytų viršūnių
    Artimiausios nenudažytos viršūnės *k* paieška
    Viršūnės *k* nagrinėjimas

(jei ieškome tik kelio iki viršūnės *t* (2-as uždavinys), tai naudojama sąlyga „Kol viršūnė t nenudažyta“)

Procedūra. Artimiausios nenudažytos viršūnės *k* paieška

min = *sum* (*min* – mažiausias rastas atstumas, pradžioje lygus begalybei)

Kiekvienai viršūnei *i* nuo 1 iki *n*

    Jei viršūnė *i* nenudažyta IR *d*[*i*] < *min*
        min = *d*[*i*]
        k = *i*

Jei *min* = *sum* (neradome kitos viršūnės, iki kurios atstumas mažesnis už begalybę; vadinasi, likusios nenudažytos viršūnės nėra pasiekiamos iš s)

    Baigiame darbą

Procedūra. Viršūnės k nagrinėjimas

Nudažome viršūnę k

Kiekvienai viršūnės k kaimynei u (egzistuoja briauna arba lankas (k, u), kurio svoris yra $c(k, u)$)

Jei viršūnė u nenudažyta IR $d[k] + c(k, u) < d[u]$
(Jei atstumas nuo s iki k + briaunos (k, u) svoris yra mažesnis už iki šiol rasto trumpiausio kelio iki u ilgį)

$d[u] = d[k] + c(k, u)$ (tai radome naują trumpiausią kelią nuo s iki u)

$prec[u] = k$ (ir šiame kelyje į u patenkame iš k)

Kodėl Dijkstros algoritmas veikia?

- Kodėl galime teigti, kad iš visų nenudažytų išsirinkus artimiausią viršūnę *k*, trumpiausias kelias iki jos jau rastas?
- Tarkime priešingai: egzistuoja dar neatrastas trumpiausias kelias iki *k*.
- Tegu kelias baigiasi briauna (lanku) (v, k).
- Tuomet *v* dar nenudažyta, nes jei ji būtų nudažyta anksčiau nei *k*, tai (v, k) jau būtume nagrinėję ir būtume radę šį trumpiausią kelią iki *k*.
- Kadangi dar nerastas kelias yra trumpesnis už iki šiol rastą, tai turi galioti nelygybė: $d[v] + c(v, k) < d[k]$.
- Tačiau algoritmas parenka $d[k]$ reikšmę taip, kad $d[k] \leq d[v]$, jei *v* nenudažyta.
- Taigi gavome prieštaravimą, ir neatrastas trumpiausias kelias iki *k* neegzistuoja.

Neigiami svoriai

- Dijkstros algoritmas skirtas grafams su neneigiamo svorio briaunomis (lankais).
- Nelygybė $d[v] + c(v, k) < d[k]$ iš praeito įrodymo galėtų būti teisinga, jei galiotų $c(v, k) < 0$.
- Tuomet egzistuotų trumpesnis kelias iki viršūnės k, einantis per viršūnę v, kurio algoritmas nesugebėtų aptikti.
- Galima Dijkstros algoritmo modifikacija, gebanti rasti trumpiausius kelius grafe, neturinčiame ciklo, kurio briaunų (lankų) svorių suma yra neigiama.

Procedūra. Viršūnės *k* nagrinėjimas. Modifikacija esant neigiamo svorio briaunoms

Nudažome viršūnę *k*

Kiekvienai viršūnės *k* kaimynei *u* (egzistuoja briauna arba lankas (*k*, *u*), kurio svoris yra *c*(*k*, *u*))

Jei ~~viršūnė *u* nenudažyta IR~~ $d[k] + c(k, u) < d[u]$
(Jei atstumas nuo *s* iki *k* + briaunos (*k*, *u*) svoris yra mažesnis už iki šiol rasto trumpiausio kelio iki *u* ilgį)

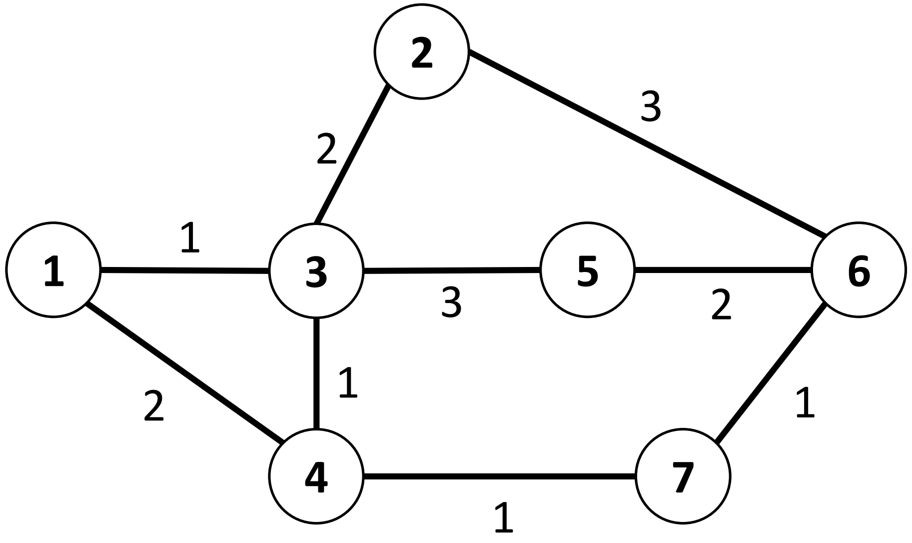
$d[u] = d[k] + c(k, u)$ (tai radome naują trumpiausią kelią nuo *s* iki *u*)

$prec[u] = k$ (ir šiame kelyje į *u* patenkame iš *k*)

Jei viršūnė *u* nudažyta

Nutriname viršūnės *u* spalvą (ji vėl tampa nenudažyta)

Užd. Duotas svorinis grafas.
Rasti trumpiausius kelius nuo pirmos iki visų likusių grafo viršūnių.
Eilės tvarka pavaizduokite pakeitimus *prec* ir *d* masyvuose, kintamojo *k* pokyčius, viršūnių dažymą.



Užd. Duotas svorinis grafas.
Rasti trumpiausius kelius nuo pirmos iki visų likusių grafo viršūnių.
Eilės tvarka pavaizduokite pakeitimus *prec* ir *d* masyvuose, kintamojo *k* pokyčius, viršūnių dažymą.