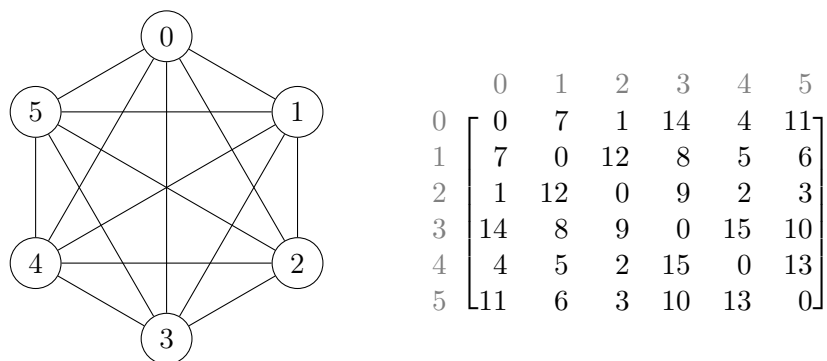


Dvanaeste vaje APS2: Problem trgovskega potnika

1 Opis problema

Pri problemu trgovskega potnika iščemo najcenejši Hamiltonov cikel na podanem uteženem neusmerjenem grafu, torej pot po grafu, ki se prične in konča v istem vozlišču, ki obišče vsako vozlišče (razen začetnega) natanko enkrat in ki ima najmanjšo skupno ceno uteži. Ponavadi predpostavimo, da je graf poln; če ni, ga lahko dopolnimo do polnega grafa tako, da uteži neobstoječih povezav nastavimo na ∞ . Prav tako bomo predpostavili, da so vozlišča oštevilčena z $0, 1, \dots, n-1$.

Na primer, v grafu na sliki 1 je pot $0 \rightarrow 2 \rightarrow 5 \rightarrow 3 \rightarrow 1 \rightarrow 4 \rightarrow 0$ s skupno ceno 31 najcenejši Hamiltonov cikel.



Slika 1: Primer grafa in pripadajoče matrike cen povezav.

2 Algoritem

Ker pot vedno pričnemo in končamo v istem vozlišču, vmes pa obiščemo vsako vozlišče natanko enkrat, lahko brez izgube splošnosti predpostavimo, da Hamiltonov cikel pričnemo in končamo v vozlišču 0. (Na primer, pot $0 \rightarrow 2 \rightarrow 5 \rightarrow 3 \rightarrow 1 \rightarrow 4 \rightarrow 0$ je enakovredna potem $2 \rightarrow 5 \rightarrow 3 \rightarrow 1 \rightarrow 4 \rightarrow 0 \rightarrow 2$, $5 \rightarrow 3 \rightarrow 1 \rightarrow 4 \rightarrow 0 \rightarrow 2 \rightarrow 5$ itd.) Število Hamiltonovih ciklov, ki se pričnejo in končajo v vozlišču 0, znaša $(n-1)!$, za izračun skupne cene vsakega cikla pa potrebujemo $O(n)$ časa, zato je časovna zahtevnost naivnega pristopa $O(n!)$. K sreči pa, kot bomo videli, lahko ta pristop nekoliko izboljšamo.

S $c(u, v)$ bomo označili ceno povezave (u, v) , z $S(u \rightarrow M)$, kjer je $M \subseteq V \setminus \{0, u\}$, pa ceno najcenejšega sprehoda, ki se prične v vozlišču u , obišče vsako vozlišče v množici M natanko enkrat in se konča v vozlišču 0. Na primer, $S(3 \rightarrow \{1, 4, 5\})$ je cena najcenejše poti od vozlišča 3 do vozlišča 0, ki vodi prek vozlišč 1, 4 in 5 (ne nujno v tem vrstnem redu). Vrednost $S(3 \rightarrow \{1, 4, 5\})$ je torej minimum cen poti $3 \rightarrow 1 \rightarrow 4 \rightarrow 5 \rightarrow 0$, $3 \rightarrow 1 \rightarrow 5 \rightarrow 4 \rightarrow 0$, $3 \rightarrow 4 \rightarrow 1 \rightarrow 5 \rightarrow 0$, $3 \rightarrow 4 \rightarrow 5 \rightarrow 1 \rightarrow 0$, $3 \rightarrow 5 \rightarrow 1 \rightarrow 4 \rightarrow 0$ in

$3 \rightarrow 5 \rightarrow 4 \rightarrow 1 \rightarrow 0$. Vrednost $S(0 \rightarrow \{1, 2, \dots, n-1\})$ je potemtakem cena najcenejšega Hamiltonovega cikla.

Bi lahko $S(u \rightarrow M)$ izračunali kako učinkoviteje kot s preverjanjem vseh $|M|!$ možnih permutacij množice M ? Premislimo. Če je $M = \emptyset$, je to kar $c(u, 0)$, saj $S(u \rightarrow \emptyset)$ predstavlja najcenejši direktni sprehod od vozlišča u do vozlišča 0. Če je $M = \{v_1, \dots, v_k\}$ za nek $k \geq 1$, pa $S(u \rightarrow M)$ izračunamo takole:

$$S(u \rightarrow M) = \min_{i=1}^k \{c(u, v_i) + S(v_i \rightarrow M \setminus \{v_i\})\}$$

Zakaj? Če želimo priti od vozlišča u do vozlišča 0 prek vseh vozlišč iz množice $M = \{v_1, \dots, v_k\}$, se najprej premaknemo v eno od vozlišč $v_i \in M$, nato pa poiščemo najcenejši sprehod od vozlišča v_i do vozlišča 0 prek vseh vozlišč iz množice $M \setminus \{v_i\}$. Med vsemi možnimi vozlišči v_i izberemo tisto, ki nam dá najmanjšo vsoto cen direktnega premika in sledečega sprehoda. Na primer, $S(3 \rightarrow \{1, 4, 5\})$ bi izračunali kot minimum vsote $c(3, 1) + S(1 \rightarrow \{4, 5\})$, vsote $c(3, 4) + S(4 \rightarrow \{1, 5\})$ in vsote $c(3, 5) + S(5 \rightarrow \{1, 4\})$.

Če ciljno vrednost $S(0 \rightarrow \{1, \dots, n-1\})$ računamo naivno, opravimo $(n-1) \cdot (n-2) \cdot \dots \cdot 2 \cdot 1 = (n-1)!$ izračunov. Vendar pa opazimo, da lahko isti izračun nastopa kot sestavni del več večjih izračunov. Na primer, pri grafu na šestih vozliščih izračun vrednosti $S(3 \rightarrow \{1, 4\})$ nastopa tako v izračunu vrednosti $S(2 \rightarrow \{1, 3, 4\})$ kot v izračunu vrednosti $S(5 \rightarrow \{1, 3, 4\})$. To pomeni, da si lahko pomagamo z dinamičnim programiranjem — bodisi tako, da ohranimo rekurzivni pristop, vendar pa vsako izračunano vrednost $S(u \rightarrow M)$ shranimo (memoiziramo), ali pa tako, da vrednosti $S(u \rightarrow M)$ računamo iterativno na podlagi »manjših«, že izračunanih vrednosti. Vlogo shrambe v obeh primerih predstavlja tabela velikosti $n \times 2^{n-1}$: prva dimenzija predstavlja vozlišče u , druga množico $M \subseteq \{1, \dots, n-1\}$, vsebina celice (u, M) pa je vrednost $S(u \rightarrow M)$. Množico M tudi tokrat predstavimo z dvojiškim številom — recimo tako, da ima vozlišče 1 največjo, vozlišče n pa najmanjšo težo. Indeks $22 = 10110_{(2)}$ tako, denimo, predstavlja množico $\{1, 3, 4\}$.

3 Primer

Oglejmo si opisani pristop na primeru grafa s slike 1. Tabela T , ki predstavlja shrambo izračunanih vrednosti, je prikazana na sliki 2. Če dinamično programiranje izvajamo po načelu od spodaj navzgor, potem tabelo polnimo od leve proti desni. Prazne celice ustrezajo neveljavnim domenam funkcije S , kljub temu pa ni nič narobe, če jih izračunamo, saj ne bodo vplivale na končni rezultat, ki se nahaja v skrajno desni celici prve vrstice.

Izračunajmo nekaj celic:

- Indeks 0 predstavlja prazno množico, zato celica $T[3][0]$ hrani vrednost $S(3 \rightarrow \emptyset)$, torej ceno najcenejše neposredne poti od vozlišča 3 do vozlišča 0. To je seveda kar vrednost $c(3, 0) = 14$.
- Indeks 1 predstavlja množico $\{5\}$, zato celica $T[4][1]$ hrani vrednost $S(4 \rightarrow \{5\})$, torej ceno najcenejše poti od vozlišča 4 do vozlišča 0, ki vodi prek vozlišča 5. Ta pot ima ceno $c(4, 5) + S(5 \rightarrow \emptyset) = c(4, 5) + T[5][0] = 13 + 11 = 24$.
- Celica $T[1][10]$ hrani vrednost $S(1 \rightarrow \{2, 4\})$, torej ceno najcenejše poti od vozlišča 1 do vozlišča 0, ki vodi prek vozlišč 2 in 4 (ne nujno v tem vrstnem redu). Ta pot ima ceno $\min\{c(1, 2) + S(2 \rightarrow \{4\}), c(1, 4) + S(4 \rightarrow \{2\})\} = \min\{c(1, 2) + T[2][2], c(1, 4) + T[4][8]\} = \min\{12 + 6, 5 + 3\} = 8$.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
1	7	17	9	23	22	29	27	35	13	10	8	15	18	22	23	27
2	1	14	6	20	23	27	28	32								
3	14	21	19	27					10	14	15	19				
4	4	24			29	36			3	16			25	29		
5	11		17		24		29		4		9		20		25	
/ 5 4 45 3 35 34 345 2 25 24 245 23 235 234 2345																
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0																31
1																
2	19	16	14	18	24	28	26	30								
3	15	23	17	25					21	18	16	23				
4	12	22			27	34			18	15			23	27		
5	13		15		25		27		19		14		24		26	
1 15 14 145 13 135 134 1345 12 125 124 1245 123 1235 1234 12345																

Slika 2: Tabela vrednosti $S(u \rightarrow M)$ za graf s slike 1. Na vrhu so zapisani indeksi, na dnu pa množice (brez zavrtih oklepajev in vejic), ki jih indeksi predstavljajo.

- Celica $T[2][22]$ hrani vrednost $S(2 \rightarrow \{1, 3, 4\})$, torej ceno najcenejše poti od vozlišča 2 do vozlišča 0, ki vodi prek vozlišč 1, 3 in 4 (ne nujno v tem vrstnem redu). Ta pot ima ceno

$$\begin{aligned}
T[2][22] &= S(2 \rightarrow \{1, 3, 4\}) \\
&= \min\{c(2, 1) + S(1 \rightarrow \{3, 4\}), c(2, 3) + S(3 \rightarrow \{1, 4\}), c(2, 4) + S(4 \rightarrow \{1, 3\})\} \\
&= \min\{c(2, 1) + T[1][6], c(2, 3) + T[3][18], c(2, 4) + T[4][20]\} \\
&= \min\{12 + 27, 9 + 17, 2 + 27\} \\
&= 26.
\end{aligned}$$

- Celica $T[0][31]$ hrani vrednost $S(0 \rightarrow \{1, 2, 3, 4, 5\})$, torej ceno najcenejše poti od vozlišča 0 do vozlišča 0, ki vodi prek vozlišč 1, 2, 3, 4 in 5 (ne nujno v tem vrstnem redu). To je seveda iskani najcenejši Hamiltonov cikel. Njegova cena se izračuna kot

$$\begin{aligned}
T[0][31] &= S(0 \rightarrow \{1, 2, 3, 4, 5\}) \\
&= \min\{c(0, 1) + S(1 \rightarrow \{2, 3, 4, 5\}), c(0, 2) + S(2 \rightarrow \{1, 3, 4, 5\}), \\
&\quad c(0, 3) + S(3 \rightarrow \{1, 2, 4, 5\}), c(0, 4) + S(4 \rightarrow \{1, 2, 3, 5\}), \\
&\quad c(0, 5) + S(5 \rightarrow \{1, 2, 3, 4\})\} \\
&= \min\{c(0, 1) + T[1][15], c(0, 2) + T[2][23], c(0, 3) + T[3][27], \\
&\quad c(0, 4) + T[4][29], c(0, 5) + T[5][30]\} \\
&= \min\{7 + 27, 1 + 30, 14 + 23, 4 + 27, 11 + 26\} \\
&= 31.
\end{aligned}$$

4 Računska zahtevnost

Prostorska zahtevnost je očitno $O(2^{n-1}n) = O(2^n n)$, časovna pa $O(2^n n^2)$, saj moramo za izračun vsake celice tabele izračunati minimum prek največ $n - 1$ drugih celic.