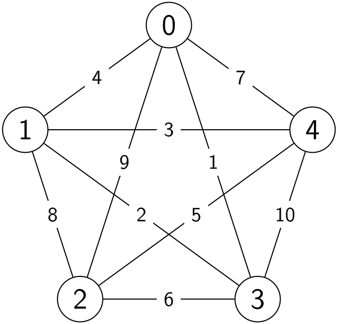
Problem trgovskega potnika

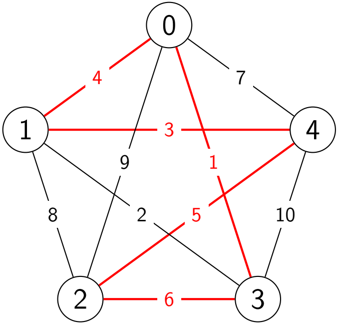
Algoritmi in podatkovne strukture 2

Problem trgovskega potnika

- Poišči najcenejši Hamiltonov cikel v polnem neusmerjenem grafu z utežmi (cenami) na povezavah
- Hamiltonov cikel
  - sprehod, ki se prične in zaključi v istem vozlišču, vsako od ostalih vozlišč pa obišče natanko enkrat
- Formalneje
  - neusmerjen graf $G = (V, E)$ z $V = \{0, 1, \ldots, n-1\}$
  - $c(u, v)$: cena povezave (u, v)
  - $C(u_1 \to u_2 \to \ldots \to u_k)$: cena sprehoda $u_1 \to u_2 \to \ldots \to u_k$
  - minimiziraj $C(\sigma(0) \to \sigma(1) \to \ldots \to \sigma(n-1) \to \sigma(0))$ preko vseh možnih permutacij σ množice $\{0, \ldots, n-1\}$
- Predpostavka
  - brez izgube splošnosti lahko vzamemo $\sigma(0) = 0$ (cikel vedno pričnemo in zaključimo v vozlišču 0)

Primer

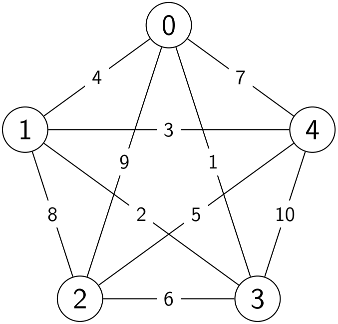




- najcenejši Hamiltonov cikel: $0 \rightarrow 1 \rightarrow 4 \rightarrow 2 \rightarrow 3 \rightarrow 0$
- cena $= 19$

Naivno reševanje

- Preizkusimo vse možne permutacije vozlišč $1, 2, \ldots, n-1$
- $O((n-1)!)$



$C(0 \to 1 \to 2 \to 3 \to 4 \to 0) = 35$

$C(0 \to 1 \to 2 \to 4 \to 3 \to 0) = 28$

$C(0 \to 1 \to 3 \to 2 \to 4 \to 0) = 24$

$\ldots$

$C(0 \to 1 \to 4 \to 2 \to 3 \to 0) = 19$

$\ldots$

$C(0 \to 4 \to 3 \to 2 \to 1 \to 0) = 35$

Boljša ideja

- Optimalni cikel $0 \rightsquigarrow 0$ zgradimo tako, da
  - pričnemo s povezavo $0 \to k$ za nek $k \in \{1, \ldots, n-1\}$
  - poiščemo optimalno pot $k \rightsquigarrow 0$
- Izberemo k, ki vodi do minimalne skupne cene povezave $0 \to k$ in poti $k \rightsquigarrow 0$

Rekurenčna formula

- Naj bo $u \in V$ in $M \subseteq V \setminus \{0, u\}$
- Naj bo $C(u \to M)$ cena optimalnega sprehoda, ki
  - se prične v vozlišču u
  - v nekem vrstnem redu obišče vsa vozlišča v M (vsako natanko enkrat)
  - se konča v vozlišču 0
- Iščemo $C(0 \to \{1, 2, \ldots, n-1\})$

Rekurenčna formula

- Izračunajmo $C(u \to M)$
- $C(u \to \emptyset) = c(u, 0)$
  - od u gremo neposredno do 0
- Če je $M = \{v_1, \ldots, v_k\}$ za $k \geq 1$, potem

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

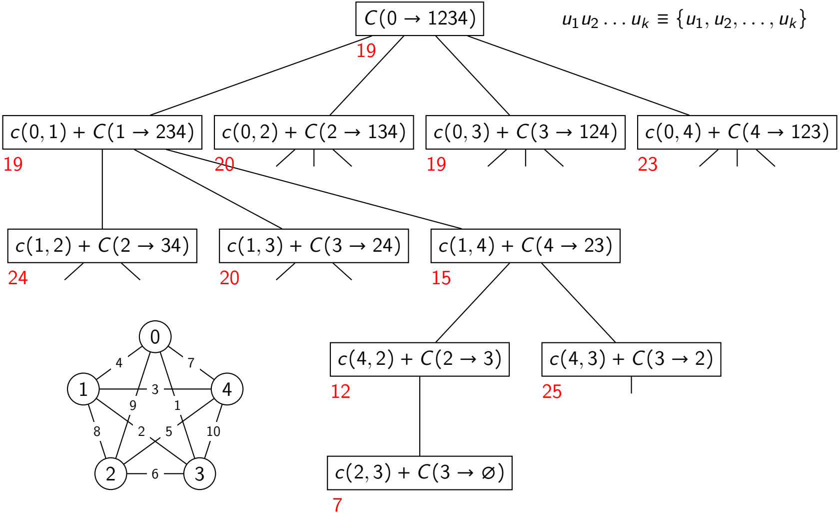
  - najprej od u neposredno do nekega $v_i \in M$
  - potem od v_i prek ostalih vozlišč iz M do 0

Primer ($n = 5$)

$$
\begin{aligned}
&C(0 \to \{1,2,3,4\}) = \min\{\\
&\qquad c(0,1) + C(1 \to \{2,3,4\}),\\
&\qquad c(0,2) + C(2 \to \{1,3,4\}),\\
&\qquad c(0,3) + C(3 \to \{1,2,4\}),\\
&\qquad c(0,4) + C(4 \to \{1,2,3\})\\
&\}\\
&C(1 \to \{2,3,4\}) = \min\{\\
&\qquad c(1,2) + C(2 \to \{3,4\}),\\
&\qquad c(1,3) + C(3 \to \{2,4\}),\\
&\qquad c(1,4) + C(4 \to \{2,3\})\\
&\}\\
&\ldots
\end{aligned}
$$

Primer

$u_1 u_2 \dots u_k \equiv \{u_1, u_2, \dots, u_k\}$



Dinamično programiranje

- Če naivno računamo po formuli, dobimo spet $O((n-1)!)$
- Lahko pa upoštevamo možnost, da se isti podproblem pojavi na več mestih v drevesu
  - npr. podproblem $C(1 \to \{3\})$ nastopa tako v izračunu $C(2 \to \{1,3,4\})$ kot v izračunu $C(4 \to \{1,2,3\})$
- Dinamično programiranje
  - vsak podproblem $C(u \to M)$ izračunamo največ enkrat

Dinamično programiranje

- Tabela D velikosti $n \times 2^{n-1}$
  - vrednost $C(u \to M)$ shranimo v celico $D[u][b(M)]$, kjer je indeks $b(M)$ vrednost bitne predstavitve podmnožice M
  - npr. vrednost $C(3 \to \{1,2,4\})$ shranimo v $D[3][13]$ $(b(\{1,2,4\}) = 1101_{(2)} = 13_{(10)})$
- Pristop od zgoraj navzdol
  - računamo po rekurenčni formuli in rezultate izračunov pomnimo v tabeli D (memoizacija)
- Pristop od spodaj navzgor
  - vrednosti $C(u \to M)$ računamo iterativno
  - tabelo D polnimo po naraščajočih $b(M)$, v okviru istega $b(M)$ pa je vrstni red lahko poljuben

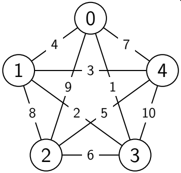
Dinamično programiranje

- Časovna zahtevnost
  - izračunamo $C(u \to M)$ za vsak $u \in \{0, \ldots, n-1\}$ in $M \subseteq V \setminus \{0, v\}$
  - $(n-1) \cdot 2^{n-2} + 1$ vrednosti
    - $C(0 \to M)$ ima smisel le pri $M = \{1, \ldots, n-1\}$
  - za vsako vrednost potrebujemo $O(n)$ časa
  - $O(2^n n^2)$
- Prostorska zahtevnost
  - tabela $n \times 2^{n-1}$
  - $O(2^n n)$

Polnjenje tabele od spodaj navzgor

smer polnjenja $\longrightarrow$

u \ M	$\emptyset$	4	3	3 4	2	2 4	2 3	2 3 4	1	1 4	1 3	1 3 4	1 2	1 2 4	1 2 3	1 2 3 4
0	0	14	2	18	18	21	16	19	8	14	7	13	21	21	19	$\boxed{19}$
1	4	10	3	14	17	17	15	15	4	10	3	9	17	17	15	15
2	9	12	7	16	9	12	7	16	12	12	11	11	12	12	11	11
3	1	17	1	17	15	18	13	18	6	12	5	12	18	18	17	17
4	7	7	11	11	14	14	12	12	7	7	6	6	17	17	16	16



$M \longrightarrow$

Sive vrednosti ne »obstajajo«,
a jih lahko brez škode izračunamo.