

Algebra a diskrétna matematika

Prehľad zo 6. týždňa

Prehľadávacie algoritmy, minimálna kostra, eulerovské grafy, hamiltonovské kružnice

Prehľadávanie grafov

Často v grafoch chceme zistiť určitú vlastnosť (napr. súvislosť, priemer, maximálny stupeň, existenciu kružníc, atď). Zo zoznamu susedov alebo matice susednosti to nie je ľahké hneď identifikovať. Mnohé takéto úlohy si preto vyžadujú efektívne a systematické preskúmanie grafu, kedy postupne navštevujeme všetky vrcholy a hrany grafu.

Uvedieme neformálny opis dvoch najviac používaných metód podľa spôsobu prehľadávania.

Prehľadávanie grafu do hĺbky (depth-first search DFS)

Hlavná myšlienka tohto prehľadávania je, že postupujeme cez susedov vrcholov tak hlboko, ako je možné.

Začneme prehľadávanie vo zvolenom vrchole, navštívime jedného jeho suseda, potom suseda tohto suseda, atď. Ak sa ďalej takto nedá pokračovať a ešte existujú nenavštívené vrcholy, tak postupujúc naspäť nájdeme prvý vrchol s ešte nenavštíveným susedom a opakujeme postup, až kým nepreskúame všetky vrcholy.

Prehľadávanie grafu do šírky (breadth-first search BFS)

V tomto prípade najprv skúmame zvolený vrchol a potom všetkých jeho susedov, potom všetkých susedov týchto susedov, atď, kým nenavštívime všetky vrcholy grafu (prípadne komponentu).

Algoritmus prechádza vrcholy podľa vzrastajúcej vzdialenosti od zvoleného vrchola. Je preto vhodný na nájdenie najkratšej cesty medzi zvoleným vrcholom a ľubovoľným iným vrcholom.

V oboch algoritmoch každá hrana dostane šípku označujúcu smer jej prvého prechodu. Hrany rozdeľujeme na stromové, ak nás dovedú k novým vrcholom, a inak na spätné.

Výstupom oboch algoritmov je (DFS alebo BFS) kostra v každom komponente súvislosti a usporiadanie vrcholov v poradí, v akom boli navštívené.

Problém minimálnej kostry: Pre súvislý graf $G = (V, E)$ s kladným ohodnotením hrán w nájdite kostru $T = (V, E')$ grafu G s najmenšou možnou hodnotou $w(T)$.

Kruskalov algoritmus: Minimálnu kostru v grafe konštruujeme postupným pridávaním hrán s najmenšou váhou tak, aby sme nevytvorili žiaden cyklus.

Vstup: Súvislý hranovo ohodnotený graf $G = (V, E)$ s n vrcholmi

Výstup: Podmnožina hrán $A \subseteq E$ taká, že graf (V, A) je minimálna kostra grafu G .

Algoritmus

- Krok 1: Polož $A = \emptyset$.
- Krok 2: Polož $F = E$.
- Krok 3: Ak $F = \emptyset$ alebo graf (V, A) je strom, ukonči, inak choď na Krok 4.
- Krok 4: Odstráň z množiny F hranu e s minimálnym ohodnotením (ak ich je viac, začni ľubovoľnou z nich). Ak graf $(V, A \cup \{e\})$ neobsahuje kružnicu, pridaj hranu e do množiny A . Choď na Krok 3.

Eulerovské grafy

Úloha: Nakreslite daný graf jedným uzavretým ťahom bez zdvihnutia tužky z papiera, pričom žiadna hrana sa neobkreslí viackrát.

Uzavretý eulerovský ťah v grafe je uzavretý sled hrán a vrcholov, v ktorom sa každá hrana vyskytuje práve raz a každý vrchol aspoň raz.

Graf je **eulerovský** práve vtedy, keď má aspoň jeden uzavretý eulerovský ťah.

Charakterizácia eulerovských grafov: Graf je eulerovský práve vtedy, keď je súvislý a všetky jeho vrcholy sú párneho stupňa.

Mostom grafu $G = (V, E)$ je taká hrana $e \in E$, pre ktorú platí, že graf $G - e$ má väčší počet komponentov ako graf G .

Tvrdenie: Ak má graf všetky vrcholy párneho stupňa, tak neobsahuje most.

Pred opisom algoritmu na generovanie eulerovských ťahov najprv opíšeme procedúru $\text{Proc}(H, u)$, ktorá v súvislom eulerovskom grafe H vygeneruje *nejaký* uzavretý ťah T začínajúci a končiaci vo vrchole u grafu H :

Proc(H, u): Vyberieme ľubovoľnú hranu $e_1 = uv_1$ v H (taká musí existovať) a vytvoríme graf H_1 vynechaním hrany e_1 z grafu H ; symbolicky, $H_1 = H - e_1$. Postup iterujeme, t.j. vyberieme ľubovoľnú hranu $e_2 = v_1v_2$ (taká musí existovať, pretože stupeň vrchola v_1 v novom grafe H_1 je nepárny – a zároveň $e_2 \neq e_1$, pretože e_1 už nie je v H_1) a vytvoríme graf $H_2 = H_1 - e_2$. Takto pokračujeme, až kým znova “narazíme” na vrchol u . To sa v niektorom kroku *musí* stať – t.j. niekde v i -tom kroku sa dostaneme do situácie, kedy v práve navštívenom vrchole v_{i-1} vyberieme hranu $e_i = v_{i-1}u$ “končiacu” vo vrchole u . Zostrojený uzavretý ťah $ue_1v_1e_2v_2 \dots v_{i-1}e_iu$ označíme symbolom T ; ten bude tvoriť výstup našej procedúry.

Algoritmus vygenerovania eulerovského ťahu v eulerovskom grafe

Vstup: eulerovský graf $G = (V, E)$

Výstup: eulerovský ťah T

Algoritmus:

- Krok 1: Polož $H = G$.
- Krok 2: Polož $T = \emptyset$.
- Krok 3: Spusť $\text{Proc}(\tilde{H}, u)$ v nejakom súvislom komponente \tilde{H} grafu H v nejakom vrchole $u \in H \cap T$ (ak $T = \emptyset$, $u \in H$), výstup označ T'
- Krok 4: Za T polož $T \cup T'$; t.j. ak $T = v_0e_1v_1 \dots e_re_rv_{r+1} \dots e_iv_0$ a $T' = uf_1 \dots f_su$, tak nové $T = v_0e_1v_1 \dots e_ruf_1 \dots f_sue_{r+1} \dots e_iv_0$.
- Krok 5: Za H polož $H - E(T')$.
- Krok 6: Ak $H = \emptyset$, ukonč, inak chod' na Krok 3.

Hovoríme, že graf G s párnym počtom vrcholov $2n$ má **perfektné párovanie**, ak G obsahuje n nezávislých hrán (žiadne 2 nemajúce spoločný vrchol).

Veta (König, 1916): Každý pravidelný bipartitný graf s aspoň jednou hranou má perfektné párovanie.

Veta (Petersen, 1891) Každý pravidelný graf stupňa 3 bez mostov má perfektné párovanie. (Tu graf nemusí byť bipartitný!)

Problém obchodného cestujúceho

Nech v grafe G vrcholy reprezentujú mestá a hrany cesty medzi mestami. Problém obchodného cestujúceho spočíva v určení trasy začínajúcej a končiacej v tom istom meste, pričom cestujúci každé iné mesto navštívi práve raz a súčet prejdenej vzdialeností (resp. nákladov, resp. časov) bude minimálny možný.

Kružnica v G je **hamiltonovská**, ak obsahuje všetky vrcholy grafu G .

Ekvivalentná formulácia problému obchodného cestujúceho: Nájsť v grafe G s kladne ohodnotenými hranami hamiltonovskú kružnicu s najmenším celkovým ohodnotením.

Algoritmy na hľadanie optimálnej hamiltonovskej kružnice sú veľmi komplikované. Nateraz sa uspokojíme s **heuristikou**, t.j. procedúrou, ktorá nájde hamiltonovskú kružnicu s “rozumne malým” ohodnotením.

Budeme uvažovať o ohodnotenom *úplnom* grafe G (aby sme sa vyhli problému existencie hamiltonovskej kružnice), pričom pre (kladné) ohodnotenie w hrán predpokladáme pre každú hranu uv **trojuholníkovú nerovnosť**:

$w(uv) \leq w(ux) + w(xv)$, pre každý vrchol x .

Heuristika zdvojenej kostry.

- Pomocou Kruskalovho algoritmu nájdeme v G najlacnejšiu kostru T .
- Na T vytvoríme uzavretý sled S , ktorý každú hranu T prejde 2-krát.
- Pomocou S budujeme kružnicu C z ľubovoľného vrchola u rekurzívne:
 1. Vydáme sa z u v smere sledu S , až kým nie sme nútení prejsť nejakou hranou v opačnom smere, t.j. keď $S = u...vwv...$. Vezmeme dočasne $C = u...vw$.
 2. Pokračujeme v traverzovaní sledu S , tentoraz z vrchola w , až po prvý výskyt nejakého vrchola x (čiže teraz $S = u...vwv...x...$), ktorý nie je v našej dočasnej C ; do C pridáme vrchol x a hranu wx . Ak taký x nie je, tak bola prejdená celá trasa S ; do C pridáme hranu wu a skončíme.
 3. Ak $S = u...vwv...xy...$ a $y \notin C$, budujeme ďalej C ako v 1 (x preberie rolu vrchola u v bode 1); ak $y \in C$ (a teda xy je prejdená druhýkrát), pokračujeme ako v 2 (x preberie rolu vrchola w v bode 2).

Z trojuholníkovej nerovnosti máme $w(C) \leq 2 \cdot w(T)$, čo považujeme za “rozumne malé”.