Algoritmy

Tomáš Matoušek

Složitost

Definice. *Časová složitost algoritmu je funkce $T : \mathbb{N} \to \mathbb{N}$, která pro danou velikost vstupu dává počet „elementárních kroků“ algoritmu, který bude na tento vstup proveden. Paměťová složitost algoritmu je funkce $S : \mathbb{N} \to \mathbb{N}$, která pro danou velikost vstupu dává „velikost paměti“, kterou algoritmus použije během výpočtu.*[1]

Algoritmy se často mezi sebou porovnávají podle toho, jakou mají složitost na velkých vstupech. Proto se zavádějí následující definice.

Definice. *Nechť $f, g : \mathbb{N} \to \mathbb{N}$. Řekneme, že f je asymptoticky menší nebo rovno g, značíme $f = O(g)$, pokud $(\exists c \in \mathbb{R}^+)(\exists n_0)(\forall n \geq n_0) : f(n) \leq cg(n)$.*

Definice. *Nechť $f, g : \mathbb{N} \to \mathbb{N}$. Řekneme, že f je asymptoticky větší nebo rovno g, značíme $f = \Omega(g)$, pokud $(\exists c \in \mathbb{R}^+)(\exists n_0)(\forall n \geq n_0) : f(n) \geq cg(n)$.*

Definice. *Nechť $f, g : \mathbb{N} \to \mathbb{N}$. Řekneme, že f je asymptoticky stejné jako g, značíme $f = \Theta(g)$, pokud $(\exists c_1, c_2 \in \mathbb{R}^+)(\exists n_0)(\forall n \geq n_0) : c_1 g(n) \leq f(n) \leq c_2 g(n)$.*

Reprezentace grafů

Definice. (seznam následníků) *Ke každému vrcholu v je dán seznam vrcholů, do nichž vede z v hrana. Je-li graf ohodnocený, pak v těchto seznamech jsou také váhy příslušných hran.*

Definice. (matice souslednosti) *Matice A typu $|V| \times |V|$, kde pro každou hranu e vedoucí z vrcholu u do v s ohodnocením $w(e)$ je $a_{uv} = w(e)$, ostatní prvky matice jsou nulové. Vrcholy u, v jsou sousední.*

Definice. (matice incidence) *Matice A typu $|V| \times |E|$, kde pro každou hranu e vedoucí z vrcholu u do v s ohodnocením $w(e)$ je $a_{ue} = w(e)$ a $a_{ve} = -w(e)$, ostatní prvky matice jsou nulové. O hraně e také říkáme, že je incidentní s vrcholy u a v.*

[1] Přesné definice těchto pojmů vyžadují formalizaci pojmu algoritmu např. prostřednictvím Turingových strojů, což zde dělat nebudeme.

Průchody grafem

Uvedeme si dva způsoby průchodu grafem: DFS (Depth First Search) a BFS (Breadth First Search). DFS prochází graf do hloubky, tj. začne s nějakým vrcholem, pak jde na jeho souseda, poté na souseda tohoto souseda atd. BFS prochází graf do šířky, nejprve projde všechny sousedy prvního vrcholu, pak všechny sousedy sousedů atd. Označíme-li $|E| = m$, $|V| = n$, pak časová složitost těchto průchodů bude $\Theta(m + n)$.

Věta. (DFS, neorientovaný graf)
Algoritmus rozdělí množinu hran grafu na dvě části: na hrany stromové a zpětné. V každé komponentě souvislosti grafu budou stromové hrany tvořit kostru této komponenty.

Příklady použití: kostra grafu, test obarvení grafu, nalezení komponent souvislosti, nalezení komponent 2-souvislosti.

Věta. (DFS, orientovaný graf) *Algoritmus rozdělí množinu hran grafu na čtyři části: na hrany stromové, zpětné, příčné a dopředné. Navíc ke každému vrcholu přiřadí dvě čísla (časová razítka) – čas vstupu do vrcholu a čas opuštění vrcholu. Pokud setřídíme vrcholy sestupně podle časů jejich opuštění, pak dostaneme topologické pořadí vrcholů.*

Příklady použití: topologické třídění, silně souvislé komponenty.

Věta. (BFS, neorientovaný i orientovaný graf) *Algoritmus rozdělí množinu vrcholů na úrovně. V i-té úrovni budou právě ty vrcholy, které mají od počátečního vrcholu, ze kterého BFS startovalo, vzdálenost i. V orientovaném grafu nám algoritmus navíc rozdělí hrany na stromové a zpětné.*

Příklady použití: nalezení nejkratších cest z daného vrcholu do jiného nebo do všech ostatních v neohodnoceném grafu (např. nalezení nejkratší cesty šachového koně z jednoho pole šachovnice na jiné).

Hladové algoritmy

Definice. (Matroid) *Matroid je dvojice (S, I), kde S je libovolná konečná neprázdná množina a $I \subseteq \mathcal{P}(S)$ je neprázdná množina s následujícími vlastnostmi:*
1. $B \in I, A \subseteq B \Rightarrow A \in I$ (dědičná vlastnost)
2. $A, B \in I, |A| < |B| \Rightarrow (\exists x \in B \setminus A) : A \cup \{x\} \in I$ (výměnná vlastnost)
Množiny v I se nazývají nezávislé podmnožiny S.

Příklad. (Grafový matroid) Nechť $G = (V, E)$ je neorientovaný graf. Pak $(E, \{A \subseteq E \mid A \text{ je les}\})$ je matroid. Nebo-li $S = E$ a množina hran A je nezávislá právě tehdy, když neobsahuje cyklus, tj. když A je les.

Věta. *Všechny maximální nezávislé množiny v matroidu mají stejnou velikost.*

Definice. (vážený matroid) *(S, I, w) je vážený matroid pokud (S, I) je matroid a $w : S \to \mathbb{R}^+$ je ohodnocení prvků S.*

Poznámka. w rozšíříme na $\mathcal{P}(S)$ tak, že $w(A) = \sum\limits_{x \in A} w(x)$ pro libovolnou $A \subseteq S$.

Věta. (hladový – greedy algoritmus) *Nechť $M = (S, I, w)$ je vážený matroid. Pak následující hladový algoritmus najde nezávislou množinu s maximálním ohodnocením:*

- $A := \emptyset$;
- *setřídíme prvky S sestupně podle jejich váhy; výsledek označme $(x_1, \ldots, x_n)$*
- *for $i := 1$ to n*
  if $A \cup x_i \in I$ then $A := A \cup \{x_i\}$;

Pozorování. Pokud algoritmus rozhodující, zda $x \in I$, má časovou složitost $\Theta(f)$, pak složitost hladového algoritmu je $\Theta(n \log n + nf(n))$, kde $n = |S|$, jelikož setřídit posloupnost čísel lze v čase $\Theta(n \log n)$.

Příklad. Úloha nalezení minimální kostry je řešitelná hladovým algoritmem na grafovém matroidu.