05. Grafy – vlastnosti

Obsah

- Základní pojmy
- Typy grafů
- Cesta a souvislost grafu
- Komponenta grafu
- Kostra grafu
- Stromy a binární stromy
- Topologické uspořádání
- Reprezentace grafů v paměti

Základní pojmy

Graf je matematická struktura tvořená množinou vrcholů (uzlů) a množinou hran, které propojují některé dvojice vrcholů.

Formálně je graf G definován jako uspořádaná dvojice G = (V, E), kde: - V je neprázdná množina vrcholů (vertices) - E je množina hran (edges), které spojují vrcholy

Základní terminologie

- **Vrchol** (uzol, node) - základní prvek grafu
- **Hrana** (edge) - spojnice mezi vrcholy
- **Stupeň vrcholu** - počet hran, které z vrcholu vycházejí
- **Sousední vrcholy** - vrcholy spojené hranou
- **Smyčka** - hrana vedoucí z vrcholu do téhož vrcholu
- **Izolovaný vrchol** - vrchol bez hran

Typy grafů

Podle orientace hran

- **Neorientovaný graf** - hrany nemají směr, vztah mezi vrcholy je symetrický
- **Orientovaný graf** (digraf) - hrany mají určený směr, vztah mezi vrcholy může být jednosměrný
  - U orientovaného grafu rozlišujeme vstupní a výstupní stupeň vrcholu

Podle ohodnocení

- **Neohodnocený graf** - hrany ani vrcholy nemají přiřazené hodnoty
- **Ohodnocený graf** (vážený graf) - hranám nebo vrcholům jsou přiřazeny číselné hodnoty (váhy)
  - Váhy mohou představovat vzdálenost, cenu, kapacitu, atd.

Další typy grafů

- **Úplný graf** - každý vrchol je spojen hranou s každým jiným vrcholem
- **Bipartitní graf** - vrcholy lze rozdělit do dvou disjunktních množin tak, že hrany vedou pouze mezi vrcholy z různých množin
- **Acyklický graf** - graf neobsahující žádný cyklus (uzavřenou cestu)
- **Multigraf** - graf umožňující více hran mezi stejnými vrcholy
- **Pseudograf** - graf umožňující smyčky a násobné hrany

Cesta a souvislost grafu

Cesta

Cesta v grafu je posloupnost vrcholů a hran, kde každá hrana spojuje předchozí a následující vrchol v posloupnosti.

Typy cest: - **Jednoduchá cesta** - žádný vrchol se v cestě neopakuje - **Kružnice (cyklus)** - cesta, která začíná a končí ve stejném vrcholu - **Hamiltonovská cesta** - cesta, která prochází všemi vrcholy grafu právě jednou - **Eulerova cesta** - cesta, která prochází každou hranou grafu právě jednou

Souvislost grafu

- **Souvislý graf** - existuje cesta mezi libovolnými dvěma vrcholy
- **Nesouvislý graf** - existují vrcholy, mezi kterými nevede žádná cesta
- **Silně souvislý orientovaný graf** - existuje orientovaná cesta mezi libovolnými dvěma vrcholy v obou směrech
- **Slabě souvislý orientovaný graf** - graf je souvislý, pokud ignorujeme orientaci hran

Komponenta grafu

Komponenta grafu je maximální souvislý podgraf původního grafu. To znamená, že: - Každé dva vrcholy v komponentě jsou spojeny cestou - Neexistuje další vrchol mimo komponentu, který by byl spojen cestou s některým vrcholem komponenty

Nesouvislý graf má více než jednu komponentu. Každá komponenta představuje izolovanou část grafu.

U orientovaných grafů rozlišujeme: - **Silně souvislé komponenty** - maximální podgrafy, v nichž existuje orientovaná cesta mezi libovolnými dvěma vrcholy - **Slabě souvislé komponenty** - maximální podgrafy, které by byly souvislé, kdyby se ignorovala orientace hran

Kostra grafu

Kostra grafu je podgraf, který: - Obsahuje všechny vrcholy původního grafu - Je souvislý - Nemá cykly (je to strom) - Má minimální možný počet hran potřebných k propojení všech vrcholů (n-1 hran pro n vrcholů)

Vlastnosti kostry: - Graf může mít více různých koster - Kostra existuje pouze pro souvislé grafy - Kostra je minimální souvislý podgraf obsahující všechny vrcholy

Minimální kostra grafu je kostra ohodnoceného grafu s minimálním součtem vah hran. Pro její nalezení se používají algoritmy: - Kruskalův algoritmus - Primův algoritmus - Borůvkův algoritmus

Stromy a binární stromy

Strom

Strom je souvislý acyklický graf. Má následující vlastnosti: - Mezi libovolnými dvěma vrcholy existuje právě jedna cesta - Má n-1 hran pro n vrcholů - Přidání jakékoliv hrany vytvoří cyklus - Odstranění jakékoliv hrany rozdělí graf na dvě komponenty

Kořenový strom je strom s jedním speciálním vrcholem označeným jako kořen. Pro kořenový strom definujeme: - **Kořen** - speciální vrchol, ze kterého vycházíme - **List** - vrchol stupně 1 (kromě kořene) - **Vnitřní vrchol** - vrchol, který není list - **Potomek** - vrchol přímo spojený s daným vrcholem směrem od kořene - **Rodič** - vrchol přímo spojený s daným vrcholem směrem ke kořeni - **Hloubka vrcholu** - vzdálenost (počet hran) od kořene k danému vrcholu - **Výška stromu** - maximální hloubka listu

Binární strom

Binární strom je kořenový strom, ve kterém má každý vrchol nejvýše dva potomky, které rozlišujeme jako levého a pravého potomka.

Typy binárních stromů: - **Úplný binární strom** - každý vnitřní vrchol má přesně dva potomky - **Vyvážený binární strom** - rozdíl výšek levého a pravého podstromu je u každého vrcholu maximálně 1 - **Binární vyhledávací strom** - pro každý vrchol platí, že všechny hodnoty v levém podstromu jsou menší než hodnota vrcholu a všechny hodnoty v pravém podstromu jsou větší

Topologické uspořádání

Topologické uspořádání je lineární uspořádání vrcholů orientovaného acyklického grafu (DAG) takové, že pokud existuje hrana z vrcholu u do vrcholu v, pak u předchází v v tomto uspořádání.

Vlastnosti: - Existuje pouze pro orientované acyklické grafy (DAG) - Graf může mít více různých topologických uspořádání - Používá se k plánování úloh s precedenčními omezeními

Algoritmy pro topologické uspořádání: - Algoritmus založený na odstranění zdrojových vrcholů (Kahn) - Algoritmus založený na prohledávání do hloubky (DFS)

Využití topologického uspořádání: - Plánování projektů (PERT, kritická cesta) - Kompilace závislostí (make, Maven) - Kurz studia (předpoklady pro předměty) - Vyhodnocení výrazů v kompilátorech

Reprezentace grafů v paměti

Existuje několik způsobů, jak reprezentovat graf v paměti počítače:

Matice sousednosti

- Čtvercová matice A o rozměrech n×n (kde n je počet vrcholů)
- A[i][j] = 1, pokud existuje hrana z vrcholu i do vrcholu j, jinak A[i][j] = 0
- U ohodnocených grafů A[i][j] = váha hrany

Vlastnosti: - Paměťová složitost: $O(n^2)$ - Časová složitost zjištění existence hrany: O(1) - Vhodné pro husté grafy - Nevhodné pro řídké grafy (plýtvání pamětí)

Seznam sousedů

- Pro každý vrchol je uchováván seznam sousedních vrcholů
- Může být implementováno jako pole seznamů nebo dynamické datové struktury

Vlastnosti: - Paměťová složitost: O(n + m), kde m je počet hran - Časová složitost zjištění existence hrany: O(stupeň vrcholu) - Vhodné pro řídké grafy - Efektivní pro procházení sousedů vrcholu

Incidenční matice

- Matice o rozměrech n×m (n vrcholů, m hran)
- Pro neorientovaný graf: A[i][j] = 1, pokud vrchol i náleží hraně j, jinak 0
- Pro orientovaný graf: A[i][j] = 1, pokud hrana j vychází z vrcholu i, A[i][j] = -1, pokud hrana j vstupuje do vrcholu i, jinak 0

Vlastnosti: - Paměťová složitost: O(n×m) - Méně často používaná než předchozí metody

Seznam hran

- Seznam všech hran grafu, kde každá hrana je reprezentována dvojicí (nebo trojicí u ohodnocených grafů) indexů vrcholů

Vlastnosti: - Paměťová složitost: O(m) - Jednoduchá implementace - Vhodné pro některé algoritmy (např. Kruskalův algoritmus) - Nevhodné pro zjišťování sousedů vrcholu

Příklady využití grafů

- **Mapy a navigace** - města jako vrcholy, silnice jako hrany
- **Sociální sítě** - lidé jako vrcholy, vztahy jako hrany
- **Počítačové sítě** - zařízení jako vrcholy, spojení jako hrany
- **Plánování projektů** - úkoly jako vrcholy, závislosti jako hrany
- **Chemické sloučeniny** - atomy jako vrcholy, vazby jako hrany
- **Webové stránky** - stránky jako vrcholy, hyperlinky jako hrany
- **Logistika** - sklady jako vrcholy, dopravní trasy jako hrany

Základní operace s grafy

- **Procházení grafu** - algoritmy BFS (prohledávání do šířky) a DFS (prohledávání do hloubky)
- **Hledání nejkratší cesty** - Dijkstrův algoritmus, Bellman-Fordův algoritmus, Floyd-Warshallův algoritmus
- **Hledání minimální kostry** - Kruskalův a Primův algoritmus
- **Hledání komponent** - pomocí BFS nebo DFS
- **Detekce cyklů** - pomocí DFS nebo disjunktních množin