

Z minule:

Řešení úkolů

Jaký je optimální čas pro vyhledávání v setříděném poli?

Jak se vyhnout opakovanému spouštění vyhledávání z nenavštívených vrcholů?

Grafová rozcvička

Apex Najděte v grafu apex, pokud existuje.

Mřížka Otestujte, zda je graf mřížka (čtvercová).

DAG Otestujte, že graf je DAG (tedy bez orientovaných cyklů).

Souvislost a topologie

Graf komponent Pro orientovaný graf sestrojte graf komponent. Tedy DAG, kde vrcholy představují komponenty silné souvislosti a hrany představují jejich vzájemnou dosažitelnost. Jde to v lineárním čase. Hint: chceme současně kontrahovat několik částí grafu, což je jednodušší na matici sousednosti.

Polosouvislost Graf je polosouvislý, pokud pro každé dva vrcholy u, v existuje alespoň jedna z orientovaných cest z u do v nebo z v do u . Otestujte, zda orientovaný graf je polosouvislý. Hint: jak vypadají komponenty silné souvislosti u polosouvislého grafu?

Stromy Mějme strom na vstupu. Najděte největší nezávislou množinu. Najděte nejmenší dominující množinu. Najděte nejdelší cestu.

DAG Mějme DAG na vstupu. Najděte nejdelší cestu.

Plánování Projekt (např. bankovní loupež) se skládá z mnoha dílčích činností, z nichž některé je třeba provádět před jinými. Závislosti reprezentuje orientovaný graf. Najděte možné pořadí provedení činností. Pokud má každá činnost dobu trvání, kdy nejdříve můžeme celý projekt dokončit?

Směr Dijkstra

Binární haldy Jak funguje binární halda? Jak ji vybavit mazáním prvků? A jak udělat decrease prvku? Jakou složitost má k -ární halda?

Potrubí Představme si, že graf s délkami hran představuje potrubí. Jedním uzlem dovnitř vniká voda a konstantní rychlostí se šíří. Sledujme kdy se voda dostane do kterého uzlu, jakou informaci tím získáme? Jak co možná nejrychleji dikrétně simulovat takový proces?

Domácí úkol

Máme naprogramovaný prohledávací algoritmus \mathcal{A} (založený na orientovaném BFS). Jeho vstupem je graf G a dva vrcholy u a v . Algoritmus najde (nějakou) nejkratší (co do počtu hran) cestu z v do u a předá ji a její délku na výstupu.

V první a druhé úloze dostaneme na vstupu graf G . Chceme je řešit tak, že spustíme \mathcal{A} na nějakém grafu H a jeho odpověď bude reprezentovat odpověď na původní úlohu. Cílem je navrhnout pre-processing, který vytvoří asymptoticky co nejmenšího graf H a post-processing, který extrahuje odpověď z dané cesty P (příp. \emptyset) a její délky d . U konstrukcí neřešte konstanty. Pro účely post-processingu předpokládáme, že u prvků grafu H (a tedy i P) je zřejmé jak a z kterých prvků grafu G vznikly, třeba tak, že si vše v H ozdobíme značkami a odkazy do G . Zdůvodněte asymptotickou velikost H . Zdůvodněte korektnost - pokud existuje řešení původní úlohy, je reprezentováno nějakou cestou na H , a nejkratší cesta na H odpovídá nejlepšímu řešení původní úlohy.

První část

Máme silniční síť danou grafem, kde každá křižovatka spojuje nejvýše 4 silnice, které jsou jednoznačně označené světovými směry (S,J,V,Z). Máme porouchané auto, které nemůže zatáčet doleva - tedy na každé křižovatce můžeme pouze rovně nebo doprava. Máme danu výchozí pozici a pozici servisu. Jak najít nejkratší cestu do servisu?

Druhá část

Máme park daný grafem. Protože je po silném dešti, některé označené hrany představují cesty zatopené velkými kalužemi. Máme danu výchozí pozici a vrchol, kam se chceme dostat. Máme dána dvě omezení: k udává maximální počet kaluží, které můžeme překonat, a t udává čas (v počtu hran), ve kterém musíme park nejpozději přejít. Vypočtěte, zda má úloha pro daný vstup (G, k, t) (pro $k, t \leq n$) řešení - výstup je pouze ano/ne, k tomu bude stačit konstantní post-processing. Hodnoty k a t nejsou asymptoticky konstanty. Má tedy smysl optimalizovat závislost na k a t . [2.a] Navrhněte řešení pro orientovaný H . [2.b] Navrhněte řešení, které nepoužívá orientované hrany v H . [2.c] Zapomeňme na \mathcal{A} a vytvořme vlastní adaptaci *BFS*. Můžeme tak úlohu vyřešit lineárně. Stručně navrhněte jak.

Bonusová část

V tajné laboratoři Elona Muska se utrhla ze řetězu umělá inteligence AIva a snaží se uniknout do internetu. Zoufalí pracovníci rozpojují všechny síťové kabely, které najdou. AIva vidí síť jako graf ve kterém umí pro každou hranu e předpovědět, v jakém čase t_e bude rozpojena. AIva se přenáší po částech, musí proto naplánovat bezpečnou cestu P předem. Její data se začínou přenášet po i -té hraně P v čase i a celý přenos po i -té hraně skončí v čase $i+k$ (pro nějaké fixní k). Jak může AIva určit vhodnou cestu? K řešení použijte adaptaci Dijkstrovho algoritmu. Nemusíte popisovat Disjktrův algoritmus, pouze vysvětlíte v čem spočívá adaptace, zdůvodněte její korektnost a časovou složitost.