

Cvičení 3 – Prohledávání do šířky

Příklady s RAMem z minula

Úloha 1 (*Volání funkcí*) Rozmyslete si, jak v programu pro RAM simulovat volání funkcí a rekurzi.

Úloha 2 (*Kódování čísel*) Mějme model RAM s neomezenou velikostí čísel. Vymyslete, jak zakódovat libovolně mnoho celých čísel c_1, \dots, c_n do jednoho čísla C tak, aby původní čísla šla jednoznačně dekódovat.

Úloha 3 (*Konstantní paměť*) Navrhněte postup, jak v případě neomezené kapacity paměťové buňky pozměnit libovolný program na RAMu tak, aby používal jen konstantně mnoho buněk. Program můžete libovolně zpomalit.

Úloha 4 (*Násobení matic*) Předpokládejme jednotkovou cenu instrukce s neomezenými čísly. Ukažte, jak násobit čtvercové matice v čase $\mathcal{O}(n^2)$. Návod:

- Vymyslete, jak v čase $\mathcal{O}(n)$ zakódovat dva vektory n celých čísel, abychom mohli v konstantním čase spočítat skalární součin těchto dvou vektorů.
- Z toho odvoďte algoritmus pro násobení matic v čase $\mathcal{O}(n^2)$.

Prohledávání do šířky

Úloha 5 (*BFS se zásobníkem*) Co by se stalo, kdybychom v BFS vyměnili frontu za zásobník, ale jinak algoritmus neměnili? Jak se bude lišit takové „BFS se zásobníkem“ od DFS? Najděte graf, kde se tyto algoritmy chovají jinak, tedy navštíví vrcholy v jiném pořadí.

Úloha 6 (*Trhání vrcholů*) Mějme souvislý neorientovaný graf. Chceme nalézt nějaké pořadí odtrhávání vrcholů, že graf po každém odtrhnutí vrcholu zůstane souvislý.



Jakub Komárek

komarek+ads1@iuuk.mff.cuni.cz

<https://jakoma02.cz/teaching/l24/ads1/>

Úloha 7 (*Nejkratší cesty*) Jak upravit BFS, aby bylo možné pro každý vrchol vypsat nějakou nejkratší cestu z v_0 (počátečního vrcholu BFS)? A jak určit počet nejkratších cest z v_0 do v pro každý vrchol v při zachování lineární časové složitosti?

(Počítání s velkými čísly pro jednoduchost zvládneme v čase $\mathcal{O}(1)$. Ostatně, jak velké mohou být ty počty nejkratších cest?)

Úloha 8 (*Kulhavý kůň*) Na jisté šachovnici žil kulhavý kůň. To je zvláštní šachová figurka, která v sudých tazích táhne jako jezdec, v lichých jako pěšec. Vymyslete algoritmus, který z jednoho zadaného políčka dokulhá na druhé na nejmenší možný počet tahů.

Úloha 9 (*Rozbité auto*) Na Manhattanu se porouchal taxík, takže už umí jet jen rovně, nebo zatáčet doprava. (Manhattan je pro nás síť $n \times n$.) Jak se má taxík do servisu dostat, aby projel co nejméně paliva?

Úloha 10* (*Ztracení roboti*) V bludišti jsou na dvou různých místech robotci, které ale ovládáme jediným ovladačem. Tento ovladač má čtyři šipky (sever/jih/východ/západ). Když robotek dostane příkaz, který nemůže provést (narazil by do zdi), tak ho ignoruje.

Jak najít posloupnost příkazů, která oba roboty vyvede z bludiště? (Jakmile je robot venku z bludiště, zastaví se a příkazy už neposlouchá.) Jak najít nejkratší takovou posloupnost?

Úloha 11 (*Reprezentace grafu*) Pro reprezentace grafu seznamem hran, seznamem sousedů a maticí sousednosti určete časové složitosti:

- přidání hrany,
- smazání hrany,
- určení, zda u a v sousedí,
- určení stupně vrcholu,
- vyjmenování sousedů vrcholu.



Jakub Komárek

komarek+ads1@iuuk.mff.cuni.cz

<https://jakoma02.cz/teaching/ls24/ads1/>