

Programování v GIS 1

10 - Algoritmy nad grafy

Michal Kačmařík

A924, tel.: 5512

e-mail: michal.kacmarik@vsb.cz

<https://www.hgf.vsb.cz/548/cs/>

- Laicky lze graf popsat jako puntíky propojené čarami
- Z matematického hlediska se jedná o strukturu, jejímiž vlastnostmi se zabývá oblast diskretní matematiky
- Definice grafu: graf je uspořádaná dvojice $G = (V, E)$, kde
 - V je množina vrcholů, a
 - E je množina hran (každou hranu lze též chápat jako podmnožinu - dvojici vrcholů)
- Více informací: přednáška *Teorie grafů* ze Základů informatiky

Hledání nejkratší cesty v grafu

- Z algoritmů pro hledání nejkratší cesty v grafu si předvedeme
 - Dijkstrův algoritmus (1958, Dijkstra)
 - A* (postupně vylepšován 1964 – 1968, P. Hart, N. Nilsson, B. Raphael)
- Dijkstrův algoritmus nelze použít pro grafy s hranou záporné hodnoty.
- A* je závislý na použité heuristické funkci.

Dijkstrův algoritmus

- Hledá nejkratší cesty z počátečního uzlu do všech ostatních uzlů ohodnoceného grafu.
- Pro implementaci se využívá *prioritní fronta*.
- Každému uzlu je přiřazena hodnota – součet vzdálenosti od aktuálního uzlu + hodnota hrany vedoucí z aktuálního uzlu do hodnoceného.
- V opakujícím se kroku přehodnocování uzlů se tato hodnota přepočte a do výsledku se vždy zařadí uzel s nejmenší hodnotou.
- Asymptotická složitost $O(E+V \cdot \log(V))$

Dijkstrův algoritmus

1. Vytvoříme pole, kde prvky budou tvořit struktury – uzel, hodnota aktuální vzdálenosti od předchozího uzlu. Výsledkem bude množina uzlů X s nalezenou nejkratší cestou
2. Inicializace:
 - a) Aktuálním uzlem je počáteční uzel A .
 - b) Množina X obsahuje uzel A .
 - c) Pro pole bude platit - hodnota uzlu $A = 0$, pro sousední uzly U hodnota = hodnota hrany (A, U) , pro nesousední uzly hodnota = nekonečno.

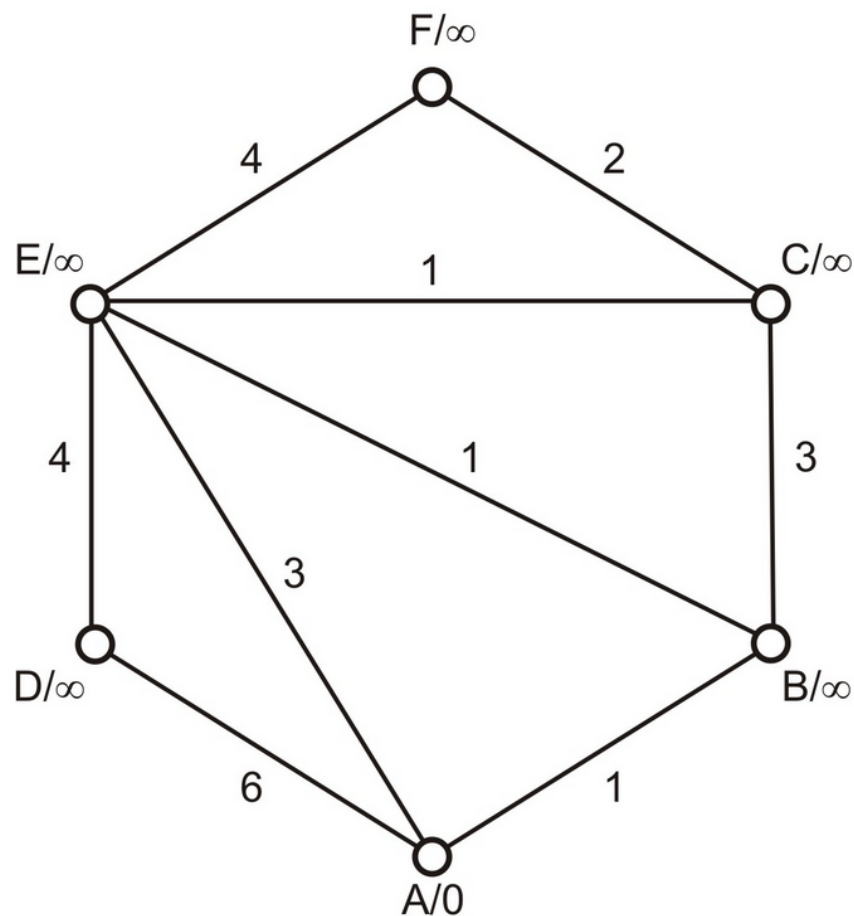
Dijkstrův algoritmus

3. Dokud množina X neobsahuje všechny uzly grafu, opakujeme:
 - a) Vybereme uzel V s nejmenší hodnotou.
 - b) Uzel V zařadíme do množiny X .
 - c) Pro každý uzel U , který sousedí s uzlem V a není prvkem množiny X , přepočteme jeho hodnotu. Pro každý uzel U vybereme minimum [stávající hodnota, hodnota uzlu V + hodnota hrany (U, V)].

Animace Dijkstrova algoritmu

Animaci lze nalézt na

<http://www.teiresias.muni.cz/amalg/www/cs/adaptation/dijkstra>



Algoritmus A*

- Grafový vyhledávací algoritmus – hledá optimální (nejkratší, nejlevnější) cestu mezi dvěma uzly v kladně ohodnoceném grafu.
- Vstup: ohodnocený graf, počáteční uzel a koncový uzel.
- Výstup: nejkratší cesta z počátečního uzlu do koncového, případně neexistence cesty.

Algoritmus A*

- Princip jako prohledávání do šířky, používá se ohodnocení uzlů pomocí speciální funkce f
- Tato funkce určuje pořadí, v kterém se mají uzly procházet
- Pro každou cestu p je funkce f definována takto:

$$f(x)=h(x)+g(x) \text{ kde}$$

- $f(x)$ – předpokládaná délka cesty x ,
- $h(x)$ – heuristická funkce pro koncový uzel cesty x (odhad hodnoty cesty z uzlu x k cíli)
- $g(x)$ – vzdálenost mezi počátečním a daným uzlem x .

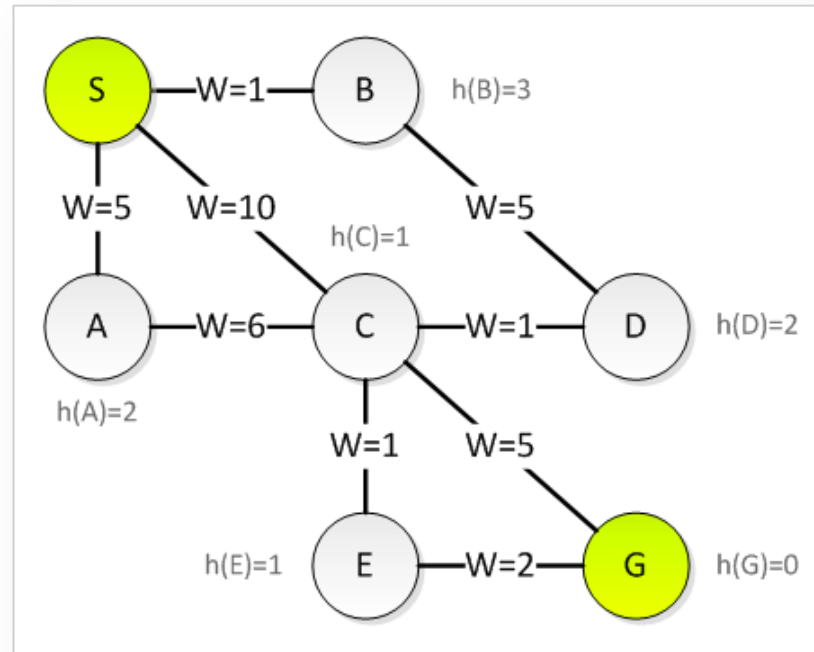
Algoritmus A*

- Volí se cesta p , která má v uzlu x hodnotu funkce $f(p)$ nejnižší (tato cesta p má nejvyšší prioritu).
- Předpokladem je, že cesty jsou bez kružnic, pro cílový uzel se v grafu nachází nanejvýš jedna cesta, a to ta nejkratší doposud nalezená.
- Funkce h musí být přípustná (nesmí nadhodnocovat vzdálenost k cíli - lze si představit heuristiku ve formě vzdálenosti vzdušnou čarou, což je v prostoru nejkratší možná cesta).
- Je-li funkce h monotónní, pak je každý uzel navštíven maximálně jednou (nedochází k návratu do již prošlého).

Algoritmus A*

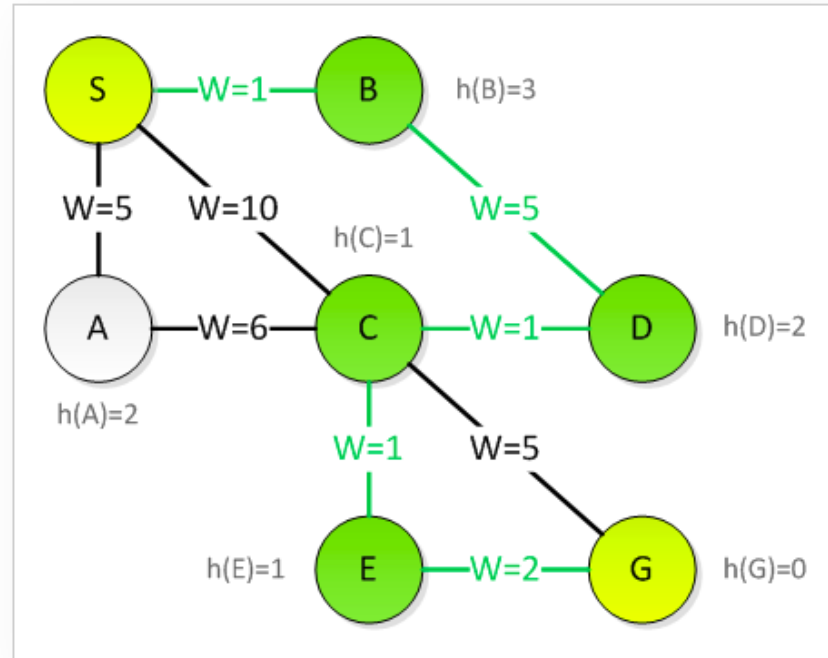
1. Vytvoříme prázdnou množinu cest F .
2. Do množiny F vložíme cestu nulové délky obsahující počáteční uzel s .
3. Dokud není množina F prázdná, opakujeme:
 - a) Z množiny F vybereme nejkratší cestu p (s nejnižší hodnotou $f(p)$) a odebereme ji.
 - b) Je-li cesta v cílovém uzlu, vrátíme ji a ukončíme výpočet.
 - c) Vytvoříme nové cesty použitím všech možných operátorů na koncový uzel cesty p , které neobsahují smyčky.
 - d) Mají-li dvě cesty konec ve stejném uzlu, odstraníme všechny, ponecháme pouze nejkratší (s nejnižší hodnotou $f(x)$).
 - e) Přidáme cestu p do množiny F .
4. Je-li množina F prázdná, pak na výstupu vypíšeme: “Neexistuje cesta z počátečního uzlu s do cílového uzlu.”

Ohodnocený graf s hodnotami heuristické funkce $h(x)$.



Algoritmus A*

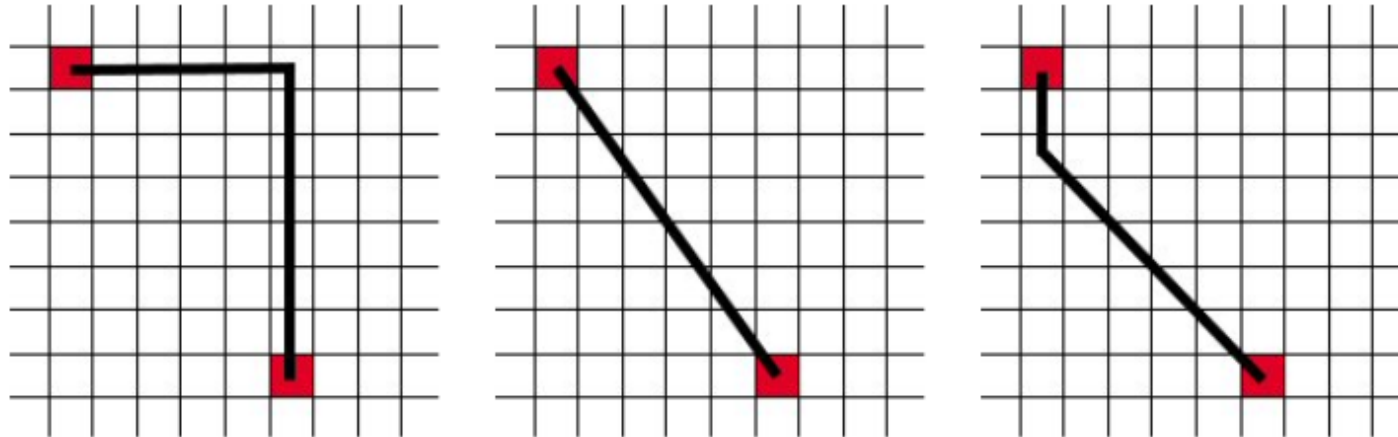
Ohodnocený graf s hodnotami heuristické funkce $h(x)$.



Algoritmus A*

Pro výpočty vzdálenosti se používá

- Manhattanská vzdálenost
- Diagonální vzdálenost
- Euklidovská vzdálenost



Složitost A*

- Závisí na odhadu vzdálenosti do cíle.
- Složitost
 - v nejhorším případě je exponenciální,
 - logaritmická pro exaktní odhady vzdálenosti.

Děkuji za pozornost

Michal Kačmařík

michal.kacmarik@vsb.cz

www.vsb.cz