

# A\* nebo take A Star

Vypracoval: Jakub Suchý

Pro predmet: Algoritmy a datove struktury I (TIN060)

## **Zadani**

Implementace algoritmu A\*, která funguje na libovolném ohodnoceném grafu v rovině -- pokud jsou váhy jedničky, bude to upravené prohledávání do šířky v lineárním case, pokud budou váhy komplikované, použijte to implementaci s k-regulární haldou pro  $k = \max(2, \text{round}(m / n))$ .

## **Co je to A\*?**

A\* je vyhledávací algoritmus, který hledá optimální cestu, je odvozen od algoritmu vyhledávání do šířky. A to tak, že používá heuristickou funkci.

V rovině je nejsnazší heuristickou funkcí vzdušná vzdálenost dvou bodů.

Pojmy: bod  $x$  vrchol rovinného grafu, funkce  $g(x)$ ,  $h(x)$  a  $f(x)$ , kde  $g(x)$  je vzdálenost od bodu  $x$  ke startu,  $h(x)$  je heuristická vzdálenost k cíli (v našem případě vzdušná vzdálenost) a  $f(x) = g(x) + h(x)$ .

## **Algoritmus**

- Vytvor prázdnou prioritní frontu vrcholů otevřených, podle  $f(x)$
- Do otevřených vlož vrchol start s počáteční cestou nulové délky
- Dokud není otevřená prázdná, opakuj:
  - Z otevřených vyber a odstran vrchol  $v$  s nejnižší hodnotou  $f(v)$  a nastav vrcholů v closed.
  - Jeli vrchol  $v$  cílovým vrcholem, zrekonstruuj a vrať cestu a skonči.
  - Projdi všechny sousední vrcholy  $x$  vrcholu  $v$ , pokud nejsou closed, spočítej  $f(x)$  a ulož je do prioritní fronty otevřených.
  - Pokud je v otevřených nějaký vrchol vícekrát, ponech ten s nejmenší hodnotou  $f(x)$  a ostatní smaž.
- Je-li prioritní fronta otevřených prázdná, vypis, že cesta neexistuje.

## **Složitost**

Pokud heuristická funkce  $h(x)$  navíc splňuje, že  $h(x) - h(y) \leq d(x, y)$ , kde  $d(x, y)$  je skutečná vzdálenost mezi body  $x$  a  $y$  (tzn. platí trojúhelníková nerovnost), pak máme zaručeno, že  $h(x)$  bude monotónní a tím pádem máme zaručeno, že každý vrchol navštívíme maximálně jednou.

Pro graf  $G=(n, m)$  v algoritmu tedy procházím všechny vrcholy v case  $O(n)$ , ale využívám k-regulární haldy jako prioritní frontu pro vrcholy s operacemi  $O(\log n)$  a nakonec procházím všechny hrany grafu v case  $O(m)$ . Z toho vyplývá, že celková časová složitost bude složitost  $O(n \cdot \log n + m)$ .