

# Intelligentné rozhodovacie systémy



## Neinformované prehl'adavanie SP

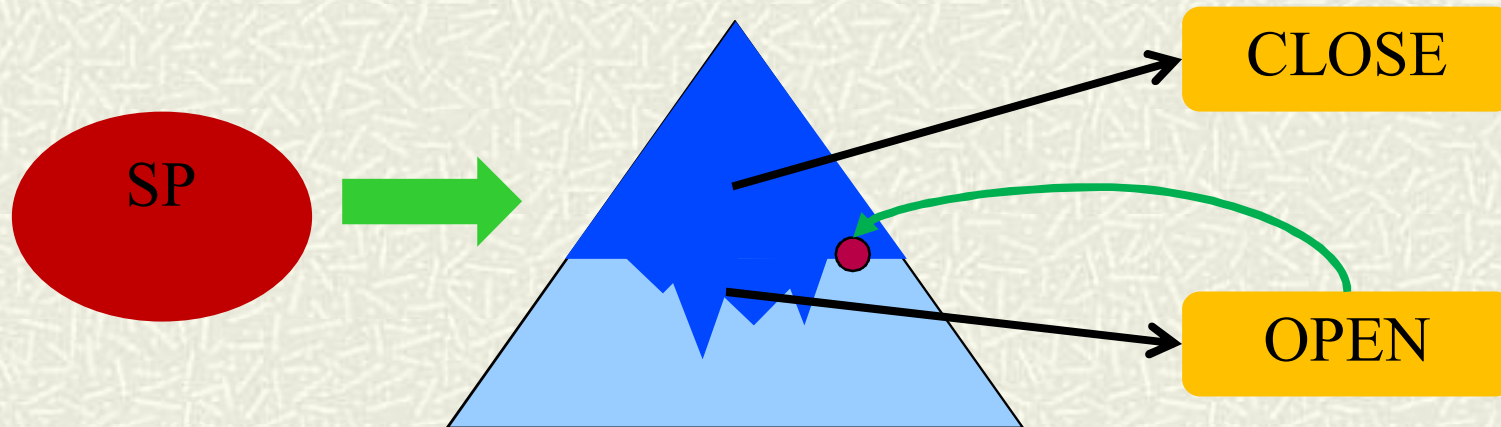
Marian.Mach@tuke.sk

<http://people.tuke.sk/marian.mach>

Október, 2018

# Princíp algoritmov

- dve dátové štruktúry
  - OPEN
  - CLOSE

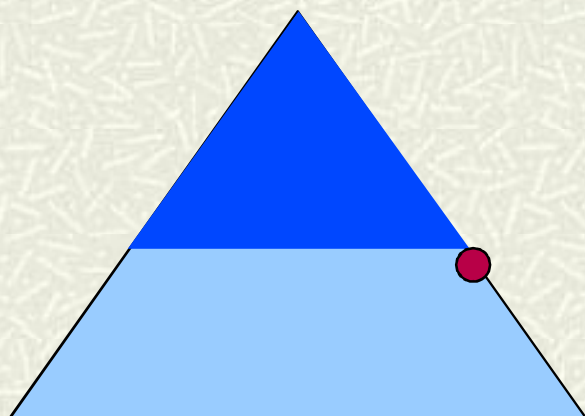


- typický obsah
  - na začiatku hľadania
  - pri neúspechu hľadania

# Algoritmus prehľadávania do šírky

- 1 Počiatočný stav  $S$  transformuj na uzol [stav= $S$ , rodič= $\emptyset$ , akcia= $\emptyset$ ] a vlož do zoznamu OPEN
- 2 Ak je OPEN prázdny, riešenie neexistuje
- 3 Vyber z OPEN uzol  $X$  ktorý je prvým uzlom zoznamu a zarad' do zoznamu CLOSE
- 4 Ak uzol  $X$  reprezentuje stav už zaradený v CLOSE, chod' na 2
- 5 Expanduj stav obsiahnutý v uzle  $X$ . Ak nemá potomkov, chod' na 2.
- 6 Ak niektorý potomok je cieľovým stavom, riešenie je nájdené (zostav ho podľa CLOSE)
- 7 Každého potomka  $Y$  stavu v uzle  $X$  transformuj na uzol [stav= $Y$ , rodič= $X$ , akcia=...] a vlož do OPEN na koniec zoznamu
- 8 Chod' na 2

## BFS - vlastnosti

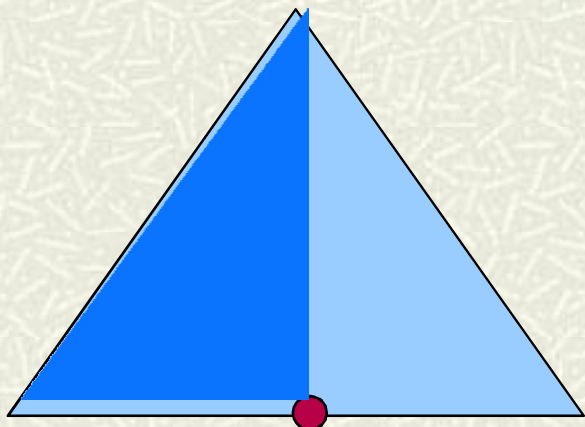


- systematické prechádzanie
- úplnosť
- optimálnosť
  - na počet použitých akcií
- časová zložitosť
  - exponenciálna
- priestorová zložitosť
  - exponenciálna

# Algoritmus prehľadávania do hĺbky

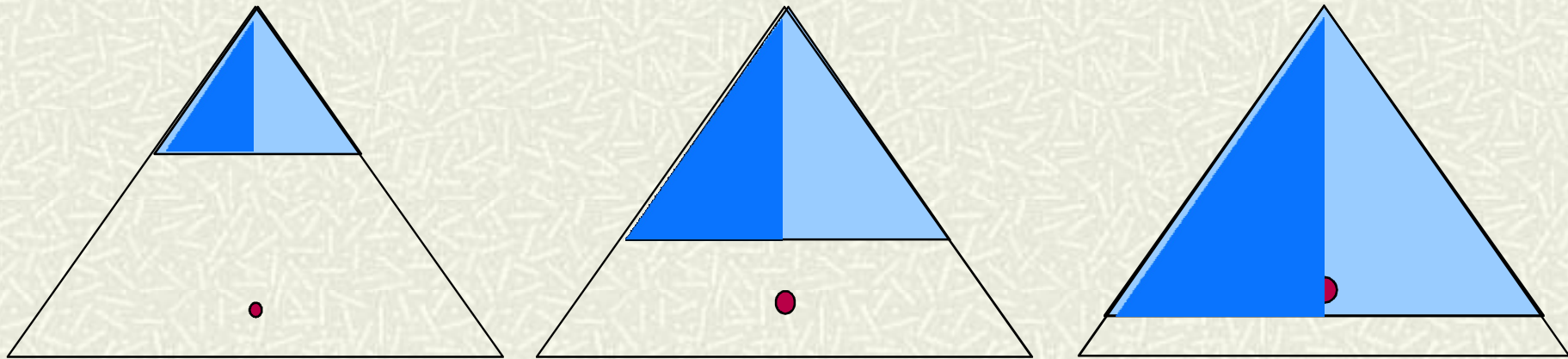
- 1 Počiatočný stav  $S$  transformuj na uzol [stav= $S$ , rodič= $\emptyset$ , akcia= $\emptyset$ ,  $h=0$ ] a vlož do zoznamu OPEN
- 2 Ak je OPEN prázdny, riešenie neexistuje
- 3 Vyber z OPEN uzol  $X$  ktorý je prvým uzlom zoznamu a zarad' do zoznamu CLOSE
- 4 Ak uzol  $X$  reprezentuje stav už zaradený v CLOSE, chod' na 2
- 5 Ak hĺbka uzla  $X$  dosiahla limit, chod' na 2
- 6 Expanduj stav obsiahnutý v uzle  $X$ . Ak nemá potomkov, chod' na 2.
- 7 Ak niektorý potomok je cieľovým stavom, riešenie je nájdené (zostav ho podľa CLOSE)
- 8 Každého potomka  $Y$  stavu v uzle  $X$  transformuj na uzol [stav= $Y$ , rodič= $X$ , akcia=...,  $h=h(X)+1$ ] a vlož do OPEN na začiatok zoznamu
- 9 Chod' na 2

## DFS - vlastnosti



- systematické prechádzanie
- úplnosť
- optimálnosť
- časová zložitosť
  - exponenciálna
- priestorová zložitosť
  - lineárna

## Iteračné prehľbovanie - vlastnosti



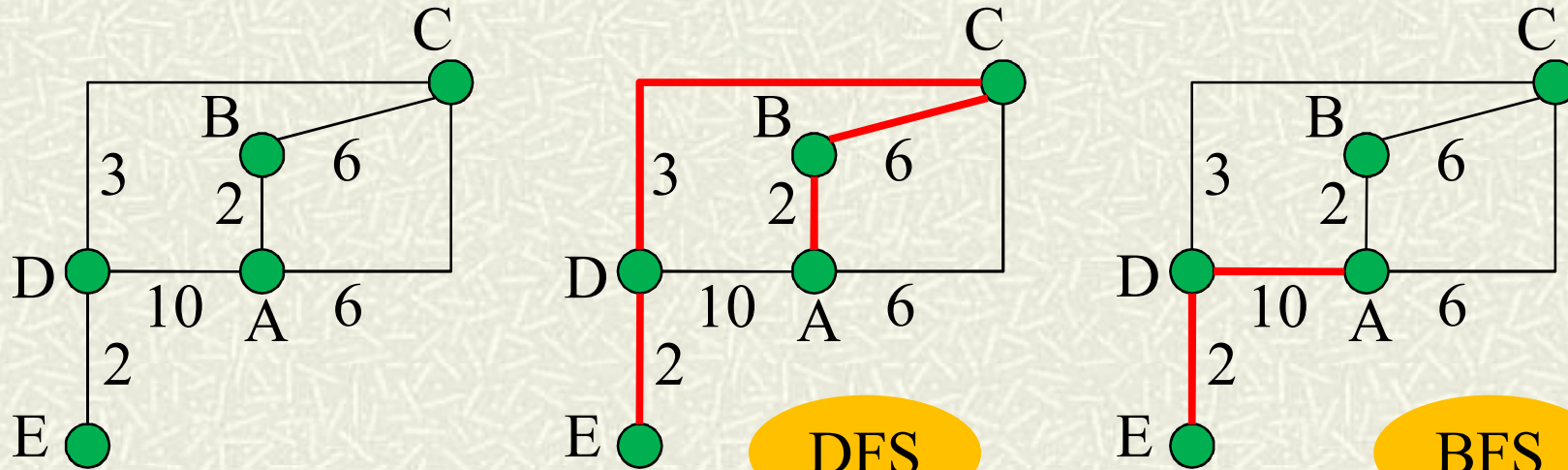
- akoby kombinoval myšlienku BFS (optimálnosť) a DFS (pamäťová nenáročnosť)
- časová zložitosť – exponenciálna
- priestorová zložitosť - lineárna

# Algoritmus iteračného prehľbovania

```
for LIMIT = 0 to  $\infty$   
  riešenie = LDFS(LIMIT)  
  if riešenie  $\neq$  null then return(riešenie) end if  
end for
```



# Použitie cenovej funkcie

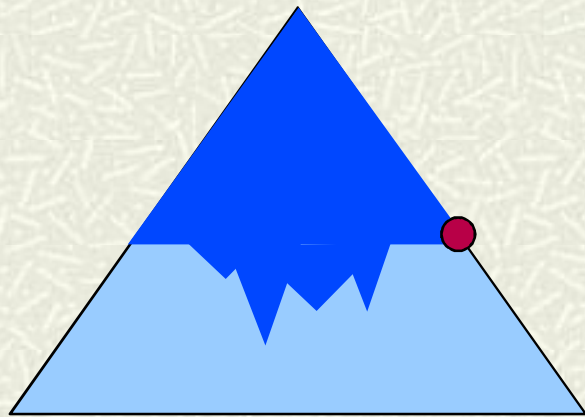


- DFS – nie je optimálne
- BFS/IS – optimálnosť iba na počet krokov
- cena
  - uniformná (konštanta)
  - kladná
  - záporná

## Prehľadávanie s uniformnou cenou

- 1 Počiatočný stav  $S$  transformuj na uzol [ $stav=S$ ,  $rodič=\emptyset$ ,  $akcia=\emptyset$ ,  $g=0$ ] a vlož do zoznamu OPEN
- 2 Ak je OPEN prázdny, riešenie neexistuje
- 3 Vyber z OPEN uzol  $X$  ktorého hodnota  $g$  je minimálna (ak viacero, tak vyber cieľový stav ak existuje, inak vyber náhodne) a zarad' do zoznamu CLOSE
- 4 Ak uzol  $X$  obsahuje cieľový stav, riešenie je nájdené (zostav ho podľa CLOSE)
- 5 Ak uzol  $X$  reprezentuje stav už zaradený v CLOSE, chod' na 2
- 6 Expanduj stav obsiahnutý v uzle  $X$ . Ak nemá potomkov, chod' na 2.
- 7 Každého potomka  $Y$  stavu v uzle  $X$  transformuj na uzol [ $stav=Y$ ,  $rodič=X$ ,  $akcia=..$ ,  $g=g(Y)+c(X,Y)$ ] a vlož do OPEN
- 8 Ak v OPEN je nejaký stav reprezentovaný viacerými uzlami, ponechaj iba ten s minimálnou hodnotou  $g$
- 9 Chod' na 2

## UC - vlastnosti



- systematické prechádzanie
  - v zmysle ceny nie polohy v strome prehľadávania
- úplnosť
- optimálnosť
  - na cenu použitých akcií
- časová/časová zložitosť
  - exponenciálna
- možná degradácia na BFS