

Neinformované a informované hľadanie

- slepé (neinformované) stratégie nevyužívajú opisy stavov na to, aby usporiadali okraj. využívajú iba polohu uzlov v strome hľadania.
- heuristické (informované) stratégie využívajú opisy stavov na to, aby usporiadali okraj. najslubnejšie uzly sa umiestnia na začiatok okraja.

1

príklad

8	2	
3	4	7
5	1	6

STAV
N₁

1	2	3
4	5	
7	8	6

STAV
N₂

pre slepú stratégiu, N₁ a N₂ sú len dva uzly (s nejakou polohou v strome hľadania)

1	2	3
4	5	6
7	8	

cieľový stav

2

príklad

8	2	
3	4	7
5	1	6

STAV
N₁

1	2	3
4	5	
7	8	6

STAV
N₂

pre heuristickú stratégiu, počítajúcu počet kameňov, ktoré nie sú na svojom mieste, N₂ je slubnejší uzol než N₁

1	2	3
4	5	6
7	8	

cieľový stav

3

poznámka

- problémy, ktoré uvažujeme, ako napr. (n^2-1) -hlavolam, sú NP-ťažké
- neočakávajme, že budeme vedieť vyriešiť ľubovoľnú (t.j. každú) inštanciu takého problému v čase lepšom než exponenciálnom
- môžeme sa usilovať vyriešiť každú inštanciu čo najefektívnejšie

→ to je účelom stratégie hľadania

4

slepé stratégie

- do šírky
 - obojsmerne
- do hĺbky
 - obmedzené
 - iteratívne sa prehľbujúce
 - do hĺbky s návratom
- rovnomerná cena (varianta do šírky)

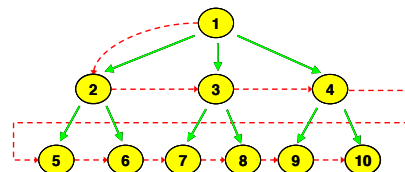
cena hrany = 1

cena hrany
= $c(\text{operátor}) \geq \epsilon > 0$

5

Hľadanie do šírky

- úplné, prípustné, exponenciálna zložitosť

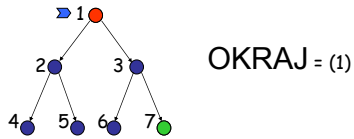


function HLADANIE-DO-ŠÍRKY(*problém*) returns *riešenie* alebo *neúspech*
return VŠEOBECNÉ-HLADANIE(*problém*, ZARAD-NA-KONIEC)

6

Hľadanie do šírky

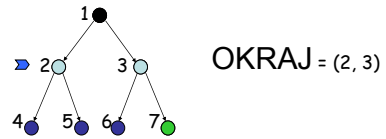
Nové uzly sa pridávajú na koniec OKRAJA



7

Hľadanie do šírky

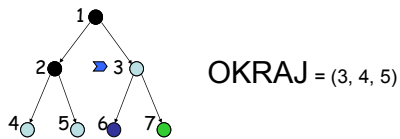
Nové uzly sa pridávajú na koniec OKRAJA



8

Hľadanie do šírky

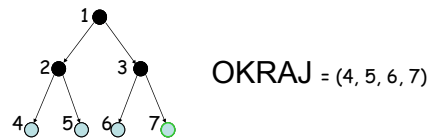
Nové uzly sa pridávajú na koniec OKRAJA



9

Hľadanie do šírky

Nové uzly sa pridávajú na koniec OKRAJA



10

Dôležité parametre

- 1) Maximálny počet nasledovníkov ktoréhokoľvek stavu
→ faktor vetvenia **b** prehľadávaného stromu
- 2) Minimálna dĺžka (\neq cena) cesty medzi počiatočným a cieľovým stavom
→ hĺbka **d** najplytšieho cieľového uzla v strome

11

Vyhodnotenie

- **b**: Vetviaci faktor
- **d**: hĺbka najplytšieho cieľového uzla
- Hľadanie do šírky je:
 - úplné
 - optimálne, ak je krok 1
- Počet vygenerovaných uzlov
???

12

Vyhodnotenie

- **b**: Vetviaci faktor
- **d**: hĺbka najplytšieho cieľového uzla
- Hľadanie do šírky je:
 - úplné
 - optimálne ak je krok 1
- Počet vygenerovaných uzlov
 $1 + b + b^2 + \dots + b^d = ???$

13

Vyhodnotenie

- **b**: Vetviaci faktor
- **d**: hĺbka najplytšieho cieľového uzla
- Hľadanie do šírky je:
 - úplné
 - optimálne ak je krok 1
- Počet vygenerovaných uzlov
 $1 + b + b^2 + \dots + b^d = (b^{d+1} - 1) / (b - 1) = O(b^d)$
- → Časová a priestorová zložitosť je $O(b^d)$

14

Časové a pamäťové nároky hľadania do šírky

Hĺbka	Počet uzlov	Čas	Pamäť
0	1	0.01 milisekundy	100 slabík
1	35	0.3 milisekundy	3.4 kilolabík
2	1225	0.01 sekundy	119 kilolabík
3	42 875	0.4 sekundy	4 megaslabík
4	1.5×10^6	15 sekúnd	143 megaslabík
5	52×10^6	8.7 minúty	4.8 gigaslabík
6	1.8×10^9	5 hodín	171 gigaslabík
7	64×10^9	7 dní	5.8 teraslabík
8	2.2×10^{12}	261 dní	204 teraslabík
9	78×10^{12}	25 rokov	7 168 teraslabík
10	2.7×10^{15}	874 rokov	250 888 teraslabík
12	3.3×10^{18}	10^6 rokov	8.7×10^6 teraslabík
...
20	7.6×10^{30}	2.4×10^{18} rokov	6.9×10^{20} teraslabík

Predpoklady: faktor vetvenia 35, 100000 uzlov / sekunda, 100 slabík / uzol

15

Poznámka

Ak problém nemá riešenie, hľadanie do šírky sa môže vykonávať *donekonečna* (ak stavový priestor je nekonečný alebo stavy môžu byť znovu navštívené ľubovoľný počet ráz)

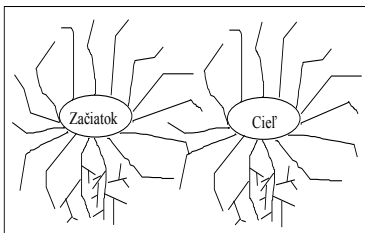
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

→

1	2	3	4
5	6	7	8
9	10	11	12
13	15	14	

16

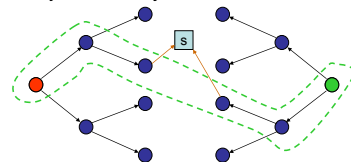
Obojsmerné hľadanie



17

Obojsmerná stratégia

fronty dvoch okrajov: OKRAJ1 a OKRAJ2

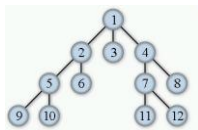


Časová a priestorová zložitosť je $O(b^{d/2}) \ll O(b^d)$ ak oba stromy majú rovnaký vetviaci faktor b

Otázka: Čo sa stane ak vetviaci faktor je rôzny od každého smeru?

18

Hľadanie do hĺbky

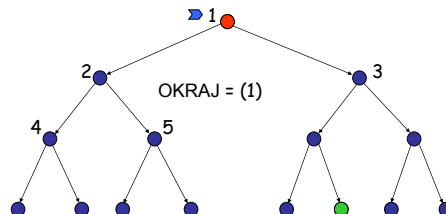


function HLADANIE-DO-HLBKY(*problém*) returns *riešenie* alebo *neúspech*
 return VŠEOBECNÉ-HLADANIE(*problém*, ZARAĎ-NA-ZAČIATOK)

19

Hľadanie do hĺbky

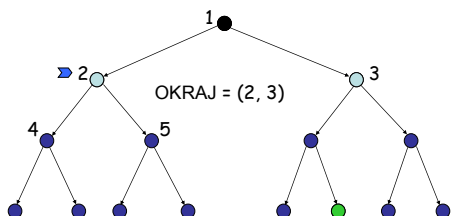
Nové uzly sa vkladajú na začiatok OKRAJA



20

Hľadanie do hĺbky

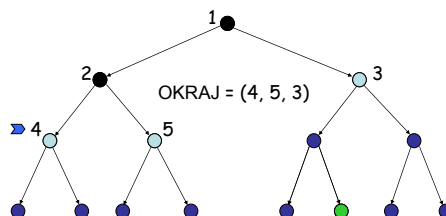
Nové uzly sa vkladajú na začiatok OKRAJA



21

Hľadanie do hĺbky

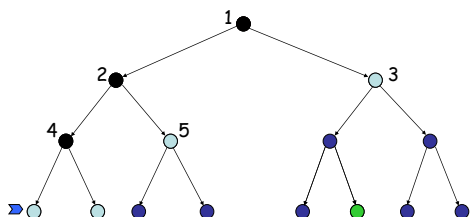
Nové uzly sa vkladajú na začiatok OKRAJA



22

Hľadanie do hĺbky

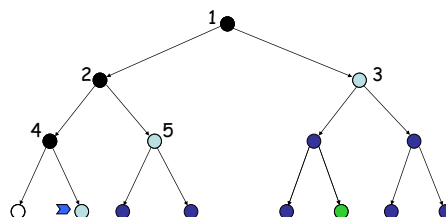
Nové uzly sa vkladajú na začiatok OKRAJA



23

Hľadanie do hĺbky

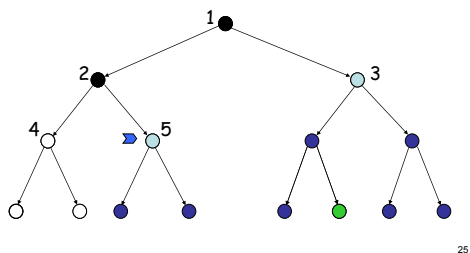
Nové uzly sa vkladajú na začiatok OKRAJA



24

Hľadanie do hĺbky

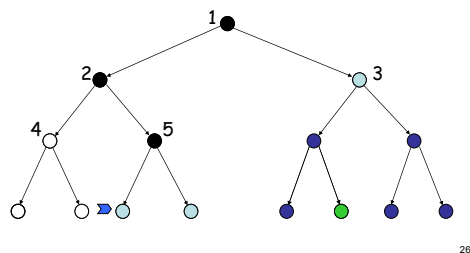
Nové uzly sa vkladajú na začiatok OKRAJa



25

Hľadanie do hĺbky

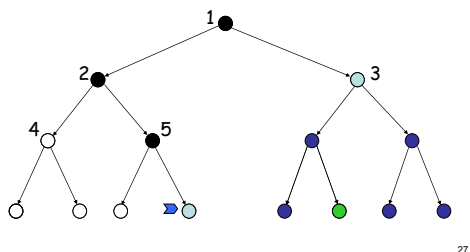
Nové uzly sa vkladajú na začiatok OKRAJa



26

Hľadanie do hĺbky

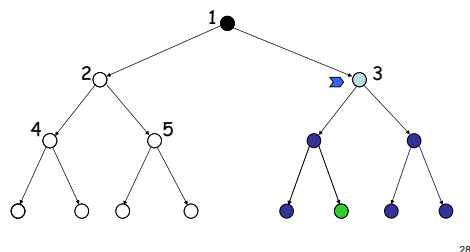
Nové uzly sa vkladajú na začiatok OKRAJa



27

Hľadanie do hĺbky

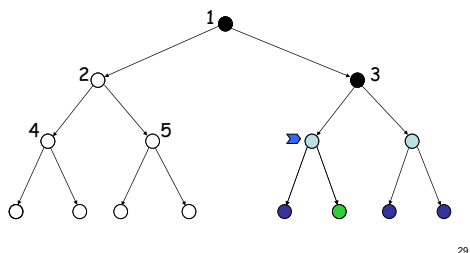
Nové uzly sa vkladajú na začiatok OKRAJa



28

Hľadanie do hĺbky

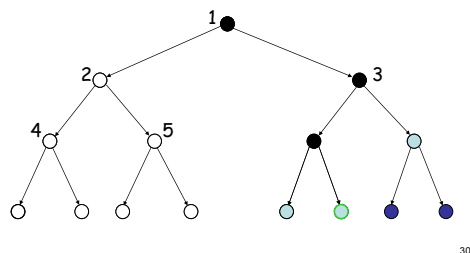
Nové uzly sa vkladajú na začiatok OKRAJa



29

Hľadanie do hĺbky

Nové uzly sa vkladajú na začiatok OKRAJa



30

Vyhodnotenie

- **b**: vetviaci faktor
- **d**: hĺbka najplytšieho cieľového uzla
- **m**: maximálna hĺbka listového uzla
- Hľadanie do hĺbky je:
 - úplné?
 - optimálne?

31

Vyhodnotenie

- **b**: vetviaci faktor
 - **d**: hĺbka najplytšieho cieľového uzla
 - **m**: maximálna hĺbka listového uzla
 - Hľadanie do hĺbky je:
 - úplné iba pre konečný strom hľadania
 - nie je optimálne
 - Počet vygenerovaných uzlov (najhorší prípad) :
 $1 + b + b^2 + \dots + b^m = O(b^m)$
 - Časová zložitosť: $O(b^m)$
 - Priestorová zložitosť: $O(bm)$ [alebo $O(m)$]
- [pripomienka: Vyhľadávanie do šírky vyžaduje $O(b^d)$ čas a pamäť]

32

Cyklicky sa prehľbujúce hľadanie

```
function CYKLICKY-SA-PREHLBUJUCE-HLADANIE(problém)
returns riešenie alebo neúspech

    for hĺbka ← 0 to ∞ do
        if OBMEDZENÉ-HLADANIE(problém, hĺbka) je úspešné
            then return jeho riešenie
    end
    return neúspech
```

33

Obmedzené prehľadávanie do hĺbky

- hľadanie do hĺbky s odsekutím v hĺbke **k**
 - hĺbka, za ktorou sa uzly nerozvíjajú
- Tri možné prípady
 - Riešenie
 - Zlyhanie – žiadne riešenie
 - Odseknutie hĺbky – nebolo by riešenia bez odseknutia

34

Cyklicky sa prehľbujúce hľadanie

Poskytuje to najlepšie z hľadania do šírky a do hĺbky

Hlavná idea: **Úplne desivé !**

IDS

Pre $k = 0, 1, 2, \dots$ do:

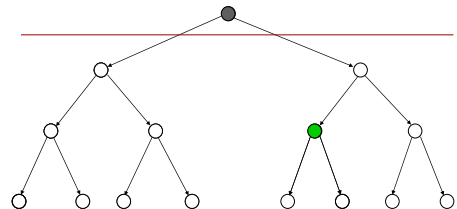
Vykonaj hľadanie do hĺbky s odsekutím v hĺbke **k**

(napr., generuj iba uzly s hĺbkou $\leq k$)

Toto sme nevedeli

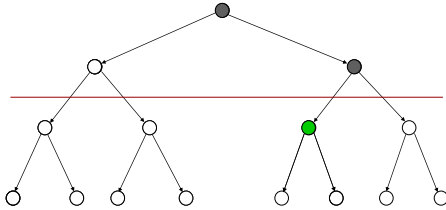
35

Cyklicky sa prehľbujúce hľadanie



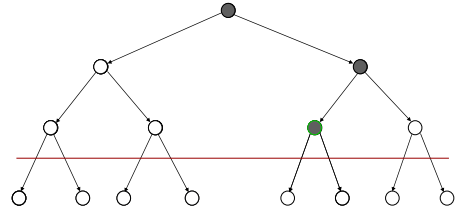
36

Cyklicky sa prehlbujúce hľadanie



37

Cyklicky sa prehlbujúce hľadanie



38

Vyhodnotenie

- Cyklicky sa prehlbujúce hľadanie je:
 - úplné
 - optimálne ak cena kroku = 1
- Časová zložitosť: $(d+1)(1) + db + (d-1)b^2 + \dots + (1)b^d = O(b^d)$
- Priestorová zložitosť: $O(bd)$ alebo $O(d)$

39

Výpočet

$$\begin{aligned}
 & db + (d-1)b^2 + \dots + (1)b^d \\
 &= b^d + 2b^{d-1} + 3b^{d-2} + \dots + db \\
 &= (1 + 2b^{-1} + 3b^{-2} + \dots + db^{-d}) \times b^d \\
 &\leq \left(\sum_{i=1, \dots, \infty} ib^{(i-1)} \right) \times b^d = b^d (b/(b-1))^2
 \end{aligned}$$

40

Počet generovaných uzlov (hľadanie do šírky a cyklické prehlbovanie)

$d = 5$ a $b = 2$

do šírky	cykl. prehľb.
1	$1 \times 6 = 6$
2	$2 \times 5 = 10$
4	$4 \times 4 = 16$
8	$8 \times 3 = 24$
16	$16 \times 2 = 32$
32	$32 \times 1 = 32$
63	120

$$120/63 \sim 2$$

41

Počet generovaných uzlov (hľadanie do šírky a cyklické prehlbovanie)

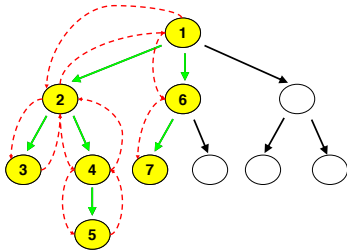
$d = 5$ a $b = 10$

do šírky	cykl. prehľb.
1	6
10	50
100	400
1,000	3,000
10,000	20,000
100,000	100,000
111,111	123,456

$$123,456/111,111 \sim 1.111$$

42

hľadanie do hĺbky s návratom



43

Hľadanie do hĺbky s návratom

```
function HLADANIE-DO-HLBKY-S-NAVRAKOM(problém)
    returns riešenie alebo neúspech
    static: front, front obsahujúci vygenerované a úplne nerozvítené uzly,
           na začiatku prázdny
           uzol, uzol stromu hľadania

    front ← VYTVOR-FRONT(VYTVOR-UZOL(ZAČIATOČNÝ-STAV[problém]))
    loop do
        if front je prázdny then return neúspech
        uzol ← SPRÍSTUPNI-PRVÝ(front)
        if CIEĽOVÝ-TEST[problém] aplikovaný na STAV(uzol, front) je úspešný
            then return VYBER-RIEŠENIE(uzol)
        if uzol má ešte nepreskúmané nasledovníky then
            front ← ZARAĎ-NA-ZAČIATOK(
                ĎALŠÍ-NASLEDOVNÍK(uzol, OPERÁTORY[problém]), front)
        else VYBER(uzol, front)
    end
```

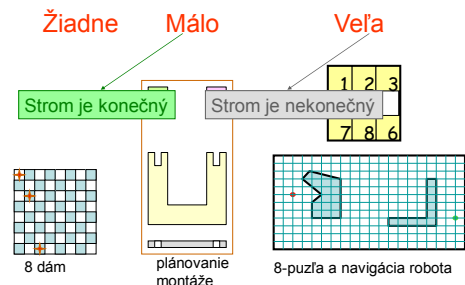
44

Porovnanie stratégií

- Hľadanie do šírky je úplné a optimálne, ale má vysokú pamäťovú zložitosť
- Hľadanie do hĺbky je pamäťovo efektívne, ale nie je úplné ani optimálne
- Cyklické prehľbovanie je úplné, optimálne, s rovnakou pamäťovou zložitosťou ako prehľadávanie do hĺbky a má skoro rovnakú časovú zložitosť ako prehľadávanie do šírky

45

Znovunavštvívené stavy



46

Vyhýbanie sa znovunavštvíveným stavom

- Vyžaduje porovnávanie opisov stavov
- Hľadanie do šírky:
 - Ulož všetky stavy združené s generovanými uzlami do NAVSTIVENE
 - Ak stav nového uzla je v NAVSTIVENE, tak zruš uzol

47

Vyhýbanie sa znovunavštvíveným stavom

- Vyžaduje porovnávanie opisov stavov
- Hľadanie do šírky:
 - Ulož všetky stavy združené s generovanými uzlami do NAVSTIVENE
 - Ak stav nového uzla je v NAVSTIVENE, tak zruš uzol

Implementovať ako rozptyľová tabuľka

48

Vyhýbanie sa znovunavštviveným stavom

hľadanie do hĺbky:

Riešenie 1:

- Ukladaj všetky stavy asociované s uzlami v aktuálnej ceste do NAVSTIVENE
- Ak stav nového uzlu je v NAVSTIVENE, tak zruš uzol
- tým sa iba vyhneme slučkám

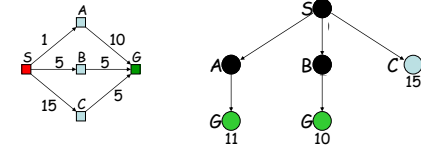
Riešenie 2:

- Ukladaj všetky generované stavy do NAVSTIVENE
- Ak stav nového uzlu je v NAVSTIVENE, tak zruš uzol
- Rovnaká pamäťová zložitosť ako pri hľadaní do šírky!

49

Stratégia rovnomernej ceny

- Každá hrana ma nejakú kladnú cenu $c \geq \varepsilon > 0$
- Cena cesty do ľubovoľného uzla N
 - $g(N) = \sum$ cien hrán pozdĺž cesty
- Cieľ je generovať cestu riešenia s minimálnou cenou
- Uzly N vo fronte OKRAJ sú usporiadané podľa stúpajúceho $g(N)$



- Nutnosť zmeniť algoritmus

50

Stratégia rovnomernej ceny

hľadanie#2

1. VLOZ(začiatkový-uzol,OKRAJ)

2. Opakuj:

- Ak prázdny(OKRAJ) tak vráť **neúspech**
- $N \leftarrow \text{VYBER}(\text{OKRAJ})$
- $s \leftarrow \text{STAV}(N)$
- Ak CIEĽ?(s) tak vráť **cestu alebo cieľový stav**.
- Pre každý stav s' v NASLEDOVNÍKY(s)
 - Vytvor uzol N' ako nasledovník N
 - VLOZ(N' ,OKRAJ)

Cieľový test sa aplikuje na uzol vtedy, keď sa tento uzol rozvinie, nie už vtedy, keď sa generuje

51

Vyhýbanie sa znovunavštviveným stavom pri stratégii rovnomernej ceny

- pre ľubovoľný stav S, keď prvý uzol N taký, že $\text{STAV}(N)=S$, sa rozvinie, cesta do N je tiež najlepšou cestou z počiatkového stavu do S
- Takže:
 - Keď uzol je rozvinutý, ulož jeho stav do ZATVORENÉ
 - Keď sa vygeneruje nový uzol N:
 - Ak $\text{STAV}(N)$ je v ZATVORENÉ, zruš N
 - Ak existuje uzol N' v OKRAJi taký, že $\text{STAV}(N') = \text{STAV}(N)$, zruš uzol N alebo N' - s najvyššou cenou cesty

52

Porovnanie neinformovaných stratégií hľadania

Kritérium	Do šírky	Rovno-mernej ceny	Do hĺbky	Obmedze-né do hĺbky	Cyklicky sa prehľbujúce	Obojsmerné
Čas	b^d	b^d	b^m	b^l	b^d	$b^{d/2}$
Pamäť	b^d	b^d	b^m	b^l	b^d	$b^{d/2}$
Pripustná?	áno	áno	nie	nie	áno	áno
Úplná?	áno	áno	nie	áno, ak $l \geq d$	áno	áno

- b je faktor vetvenia,
- d je hĺbka riešenia,
- m je maximálna hĺbka stromu hľadania,
- l je hraničná hĺbka (pri obmedzenom hľadaní do hĺbky)

53