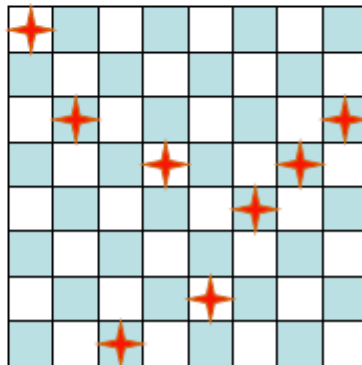
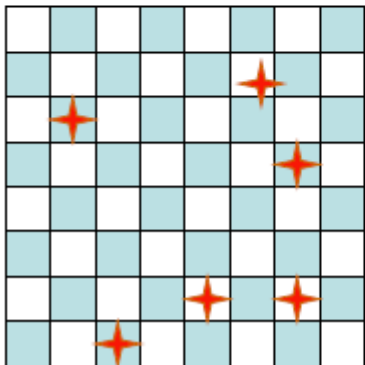
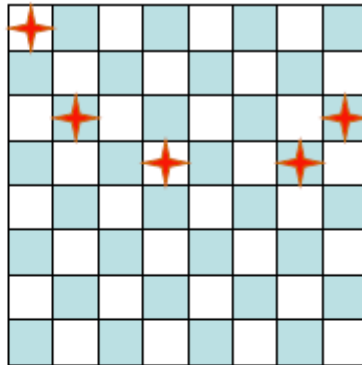
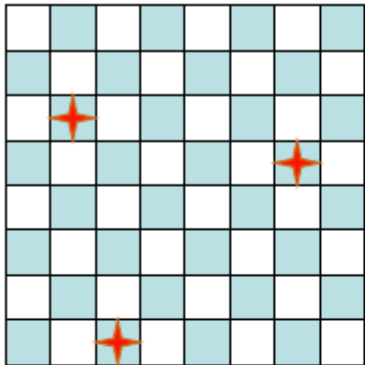




UMELÁ INTELIGENCIA

Hračkové problémy – 8 dām (1. formulácia)

2

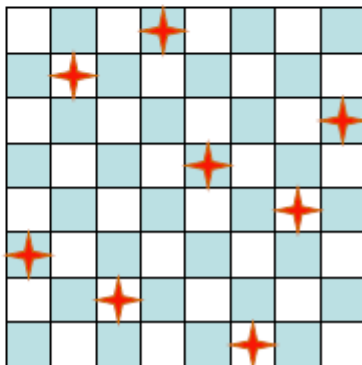
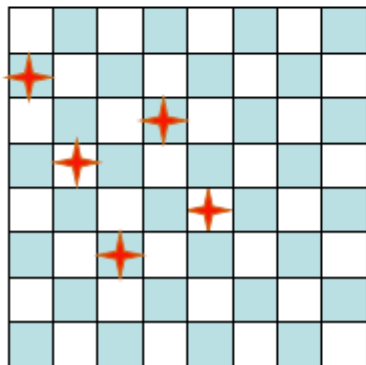
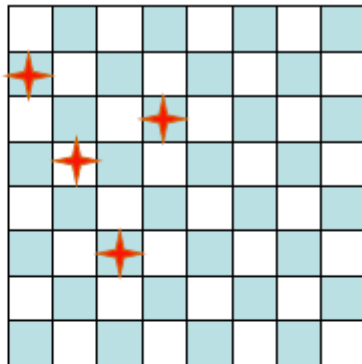
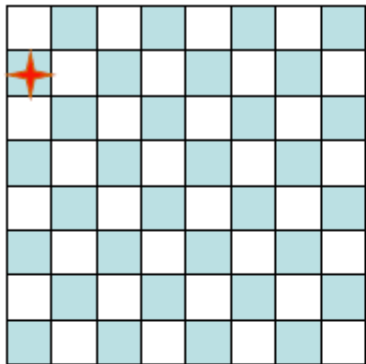


- **Stavy:** všetky možné konfigurácie ľubovoľného možného (0-8) počtu dām na šachovnici
- **Počiatočný stav:** žiadna dáma na šachovnici
- **Operátory:** polozenie dámy na ľubovoľné políčko šachovnice
- **Cena cesty:** 0
- **Cieľový test:** 8 dām na šachovnici umiestnených tak, že sa navzájom neohrozujú

$64 \times 63 \times \dots \times 57 \sim 3 \times 10^{14}$ stavov

Hračkové problémy – 8 dām (2. formulácia)

3



- **Stavy:** všetky možné konfigurácie ľubovoľného možného (0-8) počtu dām na šachovnici také, že ani jedna z dām nie je ohrozená
- **Počiatočný stav:** žiadna dáma na šachovnici
- **Operátory:** polozenie dámy na ľubovoľné políčko v najľavejšom prázdnom stĺpci také, že ju na ňom neohrozuje žiadna iná dáma
- **Cena cesty:** 0
- **Cieľový test:** 8 dām na šachovnici umiestnených tak, že sa navzájom neohrozujú

2057 stavov

Charakteristiky problémov

4

- Riešením problému je stav alebo cesta
- Problém rozložiteľný na samostatne riešiteľné podproblémy
- Problémy s ignorovateľnými krokmi riešenia
- Problémy s odčiniteľnými krokmi riešenia
- Problémy s neodčiniteľnými krokmi riešenia

Hľadanie riešenia

5

- Hľadanie riešenia je prístup k riešeniu problémov, pri ktorom nevychádzame z algoritmu riešenia problému.
- Bud' ho nepoznáme (možno preto, že ani neexistuje), alebo ho poznáme, ale pre svoju neefektívnosť je prakticky nepoužiteľný. Namiesto toho vychádzame z algoritmu, ako riešenie hľadať.

Hľadanie riešenia – algoritmus

6

function VŠEOBECNÉ-HĽADANIE(*problém*, *stratégia*)

returns *riešenie* alebo neúspech

inicializuj strom hľadania použitím začiatočného stavu z *problém*

loop do

if nie sú nerozvité uzly **then return** neúspech

vyber list za uzol na rozvitie podľa *stratégia*

if uzol predstavuje cieľový stav **then return** zodpovedajúce *riešenie*

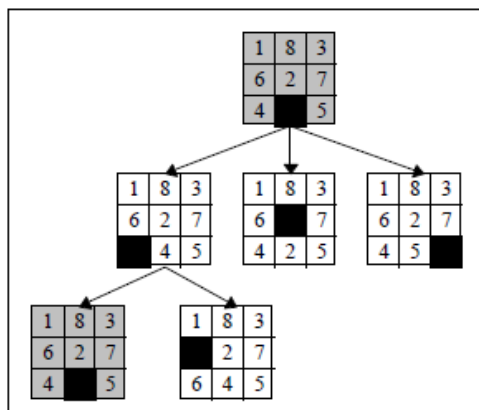
else rozvi uzol a pripíš vygenerované uzly do stromu hľadania

end

Stavový priestor a graf (strom) hľadania

7

■ Reprezentácia uzla:



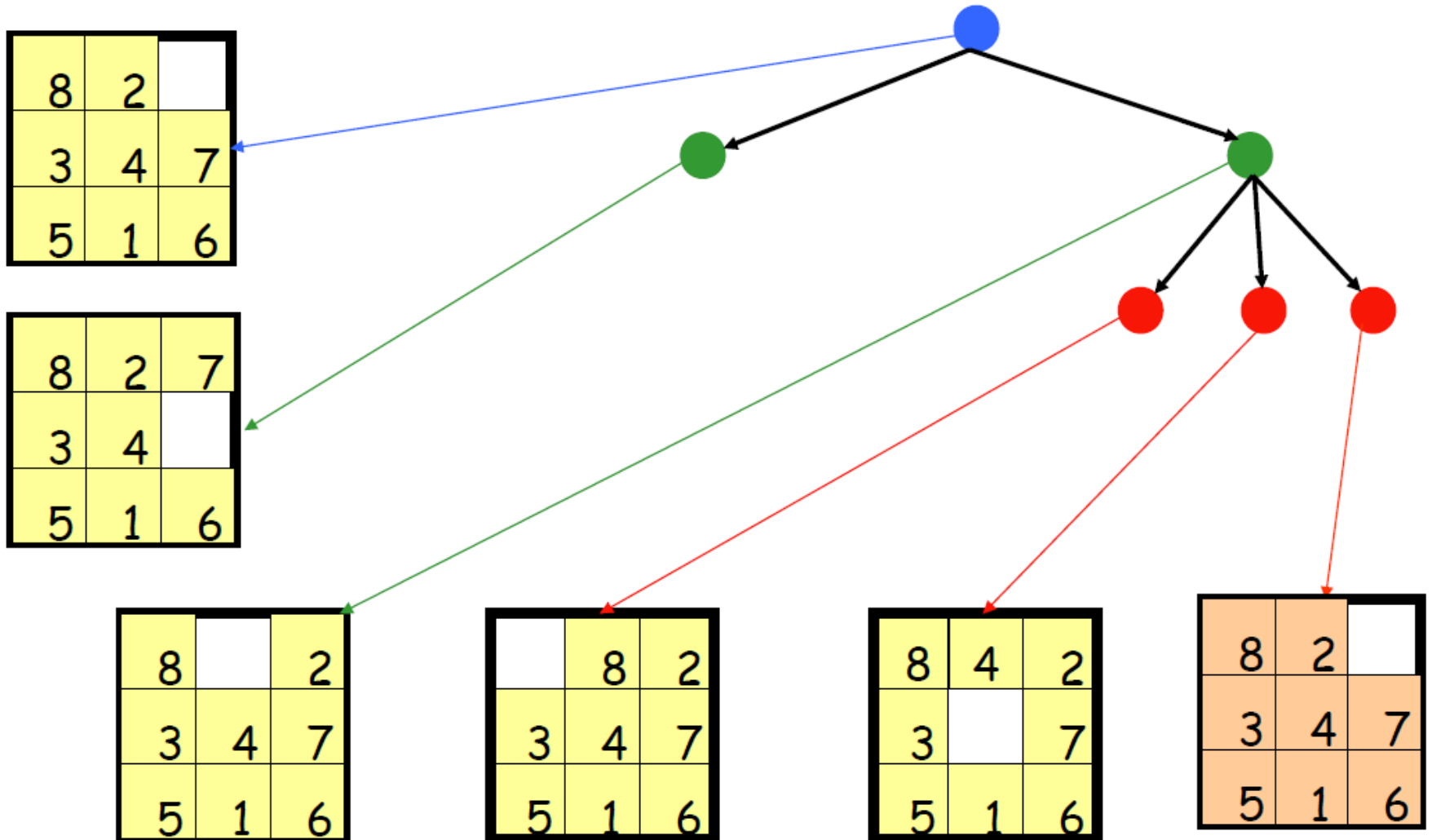
- zodpovedajúci stav zo stavového priestoru,
- uzol v strome hľadania, z ktorého sa daný uzol vygeneroval,
- operátor, ktorý sa aplikoval pri generovaní uzla (na rodičovský uzol),
- počet uzlov na ceste z koreňa do daného uzla (hĺbka uzla),
- cena cesty zo začiatočného uzla do daného uzla.

datatype UZOL

components: STAV, RODIČOVSKÝ-UZOL, OPERÁTOR,
HĹBKA, CENA-CESTY

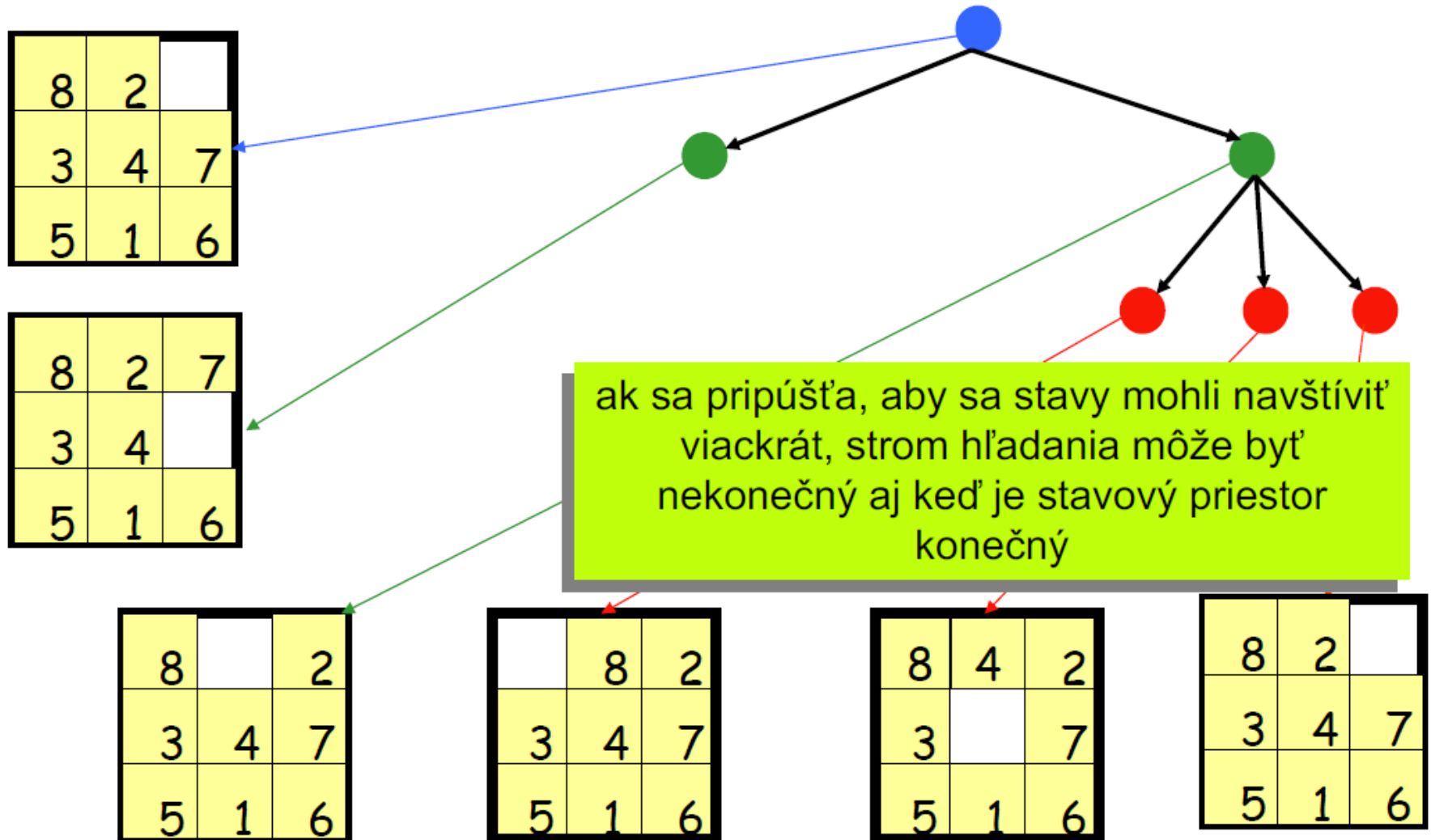
Uzly v strome hl'adania a stavy

8



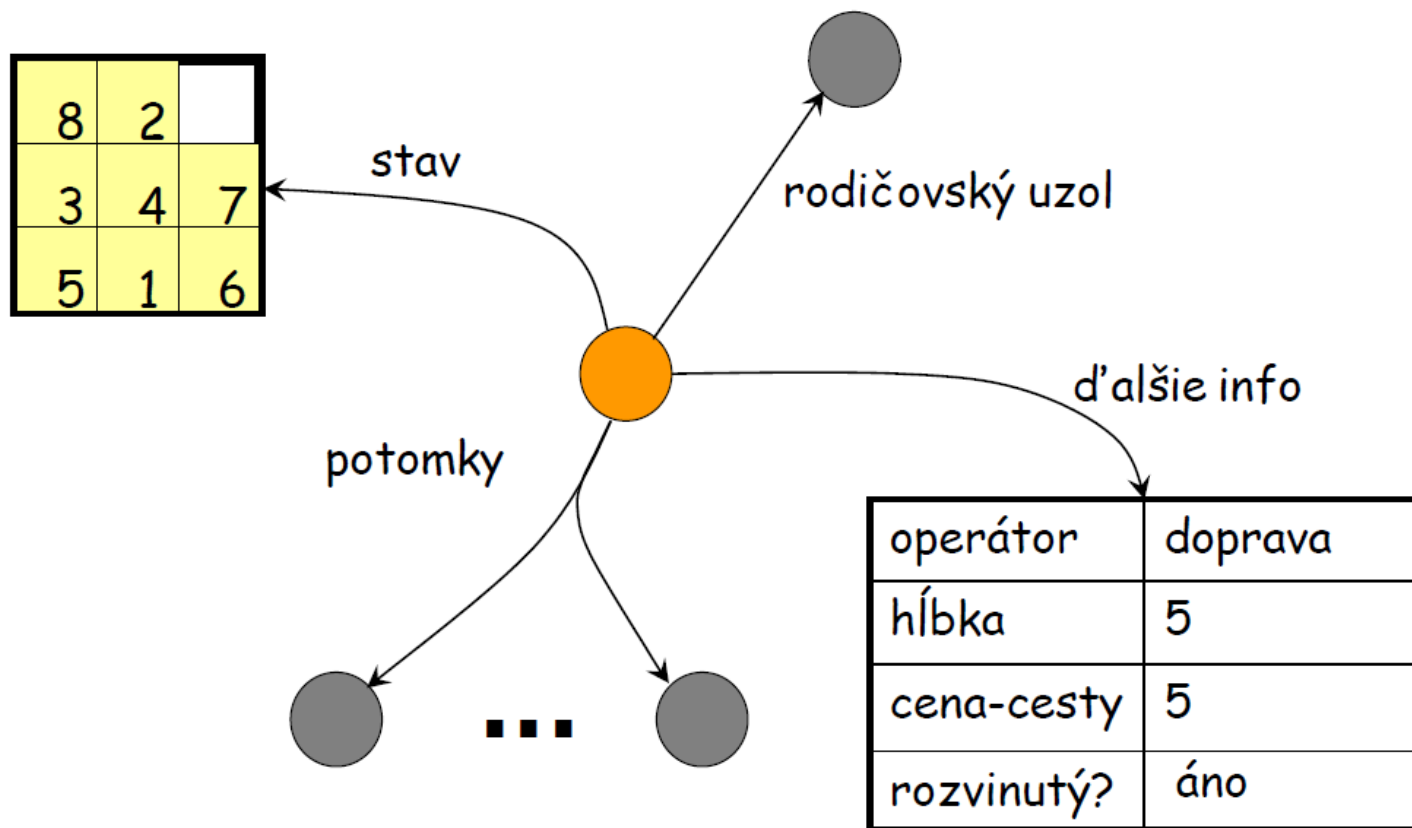
Uzly v strome hľadania a stavy

9



Dátová štruktúra pre uzol

10



hĺbka uzla N
= dĺžka cesty z koreňa do N
(hĺbka koreňa = 0)

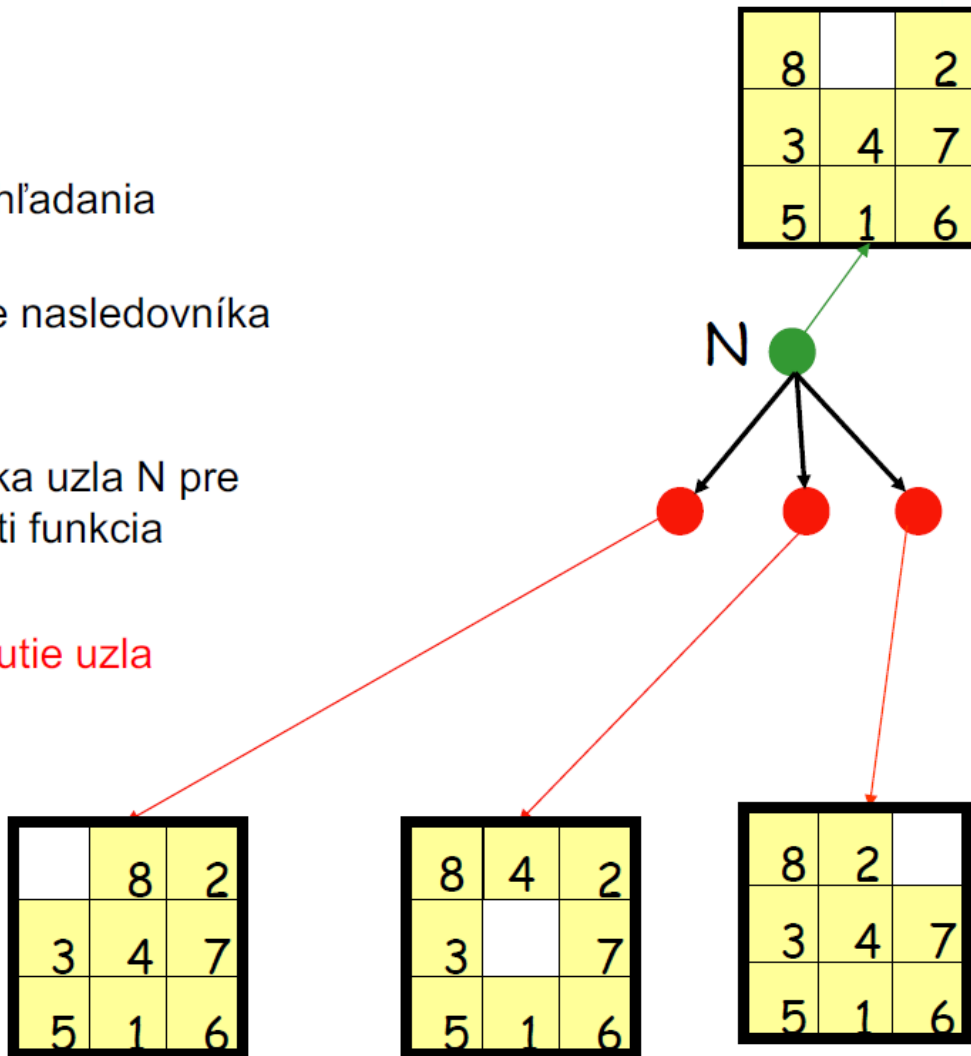
Rozvinutie uzla

11

rozvinutie uzla N v strome hľadania
pozostáva z:

- 1) vyhodnotenia funkcie nasledovníka
na $STAV(N)$
- 2) vygenerovania
potomka/nasledovníka uzla N pre
každý stav, ktorý vráti funkcia
nasledovníka

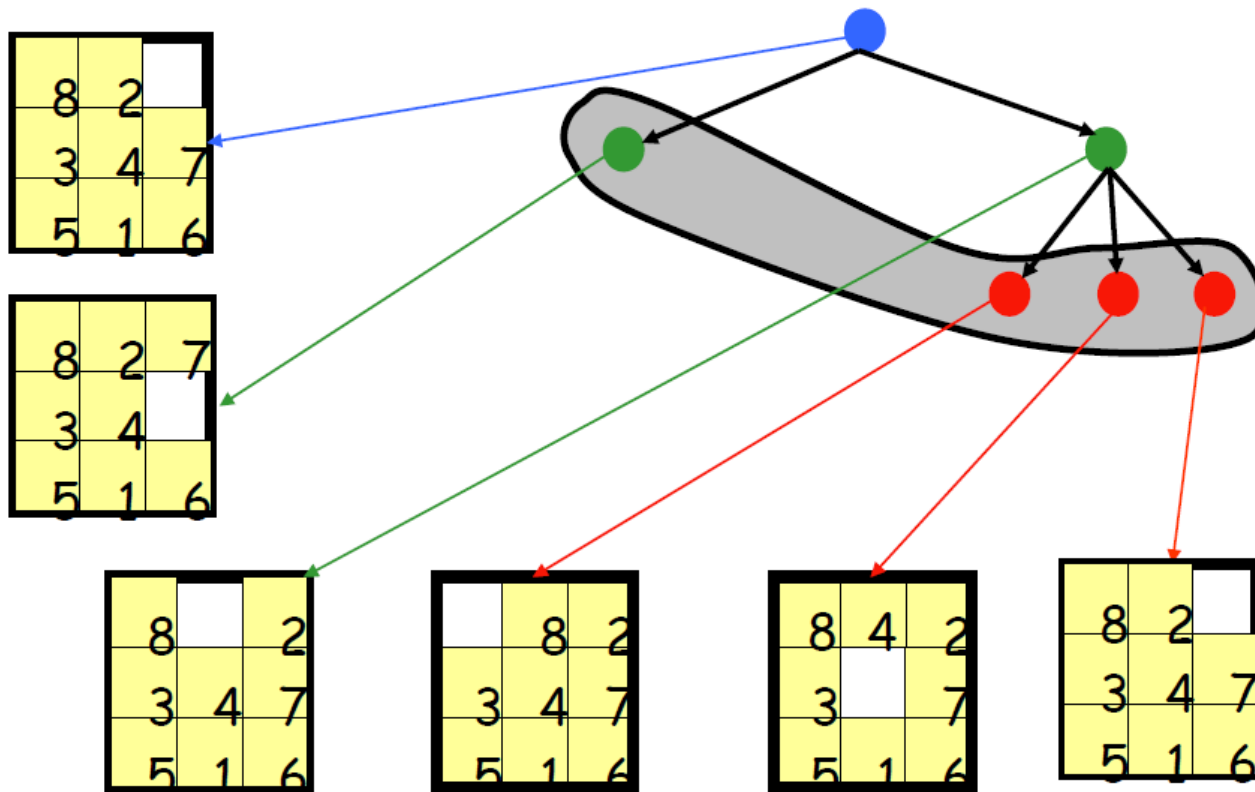
generovanie uzla \neq rozvinutie uzla



Okraj stromu hľadania

12

- okraj je množina všetkých uzlov (v strome hľadania), ktoré ešte nie sú rozvinuté



Je okraj
totožný s
množinou
listov?

Front – štruktúra na zápis množiny uzlov

13

- Angl. Queue
- Nad frontom definujeme tieto operácie:
 - ▣ VYTVOR-FRONT(prvky) vytvorí front s danými prvkami
 - ▣ PRÁZDNY(front) vráti true práve vtedy, ak front neobsahuje žiadne prvky
 - ▣ VYBER(front) odstráni prvok z frontu a vráti ho (prvok)
 - ▣ ZARAĎ-DO-FRONTU(prvky, front) vráti front po zaradení prvkov do pôvodného frontu. Rôzne druhy tejto funkcie určujú rôzne algoritmy hľadania

Všeobecný algoritmus hľadania

14

```
function VŠEOBECNÉ-HĽADANIE(problém, ZARAĎ-DO-FRONTU)
  returns riešenie alebo neúspech
  static: front, front obsahujúci vygenerované a nerozvité 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 ← VYBER(front)
    if CIEĽOVÝ-TEST[problém] aplikovaný na STAV(uzol) je úspešný then
      return VYBER-RIEŠENIE(uzol)
    front ← ZARAĎ-DO-FRONTU(ROZVI(uzol, OPERÁTOR[problém]), front)
  end
```

Stratégie hľadania

15

□ **Neinformované (slepé)**

- nemajú k dispozícii nejakú doplňujúcu informáciu o probléme
- poradie generovania stavov závisí iba od informácií získaných hľadaním a nie je ovplyvnené ani nepreskúmanou časťou grafu ani vlastnosťami cieľového stavu

□ **Informované (heuristické)**

- majú k dispozícii nejakú doplňujúcu informáciu o probléme
- heuristická informácia sa často využíva na to, aby sa zvýšila efektívnosť hľadania (t.j. znížila časová a/alebo pamäťová zložitosť) aj za cenu, že nebude dodržané ďalšie kritérium, a síce prípustnosť a/alebo úplnosť

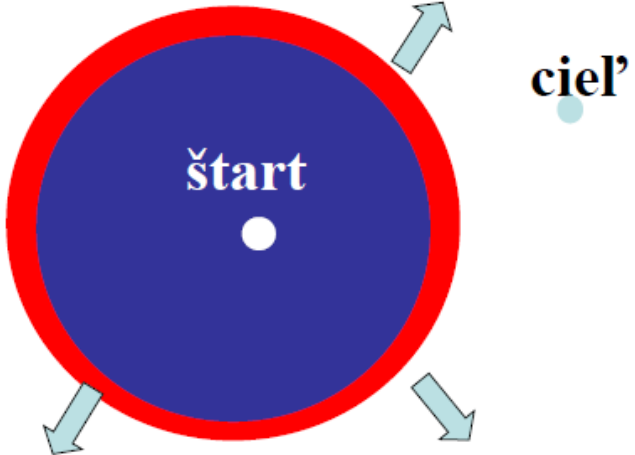
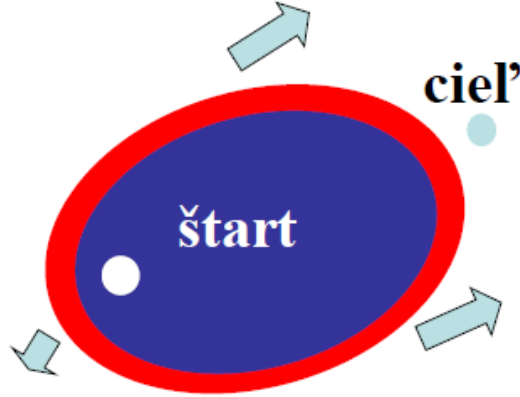
Heuristika

16

- ηύρηκα [heuréka] = našiel (objavil) som to – Archimedes
- ηύρίσκω = nájsť, objaviť
- Spôsob riešenia problému, pre ktorý nemáme algoritmus alebo presný postup ♢ heuristické riešenie problémov
- Polya: Ako to vyriešiť. 1954:
 - ▣ ak nerozumiete riešenému problému, skúste si ho nakresliť
 - ▣ ak neviete nájsť riešenie, predstavte si, že ho máte a pozrite sa, či z neho neviete odvodiť postup (pracovať odzadu)
 - ▣ ak je problém abstraktný, skúste najprv riešiť konkrétny príklad
 - ▣ skúste najprv riešiť všeobecnejší problém (paradox vynálezcu: ambicióznejší plán môže mať lepšie vyhliadky na jeho vyriešenie)
- Heuristika v informatike: postup, ktorý zvyčajne vedie k dobrému riešeniu, avšak nezaručuje, že sa nájde najlepšie riešenie, ani že sa nájde v krátkom čase, ani že sa vôbec nájde.

Stratégie hľadania

17

Neinformované hľadanie	Informované hľadanie
 <p>The diagram illustrates uninformed search. A large blue circle represents the search space, with a red border. A white dot in the center is labeled 'štart'. Three light blue arrows point outwards from the red border, indicating a uniform expansion from the start point. A small blue dot to the right of the circle is labeled 'cieľ'.</p>	 <p>The diagram illustrates informed search. An oval-shaped blue region represents the search space, with a red border. A white dot inside is labeled 'štart'. Three light blue arrows point outwards from the red border, indicating a non-uniform expansion based on some heuristic. A small blue dot to the right of the oval is labeled 'cieľ'.</p>

Stratégie hľadania

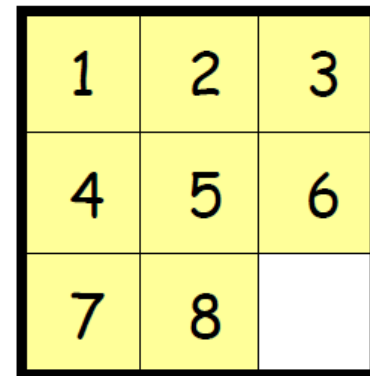
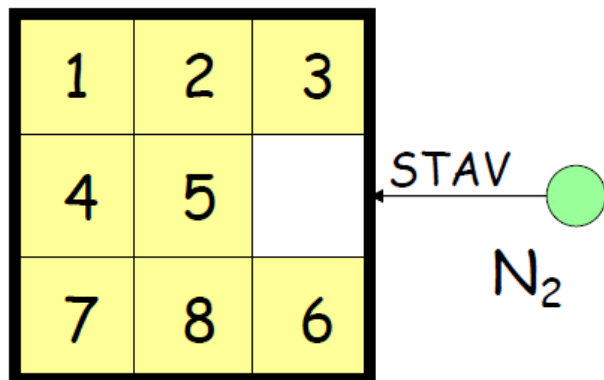
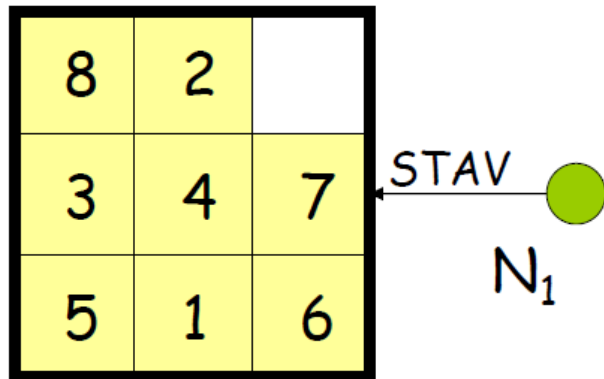
18

- ❑ **Úplnosť** – zaručuje hľadanie s danou stratégiou, že sa nájde riešenie, ak existuje?
- ❑ **Časová zložitosť** – ako dlho trvá, kým sa nájde riešenie?
- ❑ **Pamäťová zložitosť** – koľko pamäti treba na vykonanie hľadania?
- ❑ **Prípustnosť (Optimálnosť)** – nájde sa pomocou danej stratégie najlepšie riešenie, ak existuje aspoň jedno riešenie?

Príklad

19

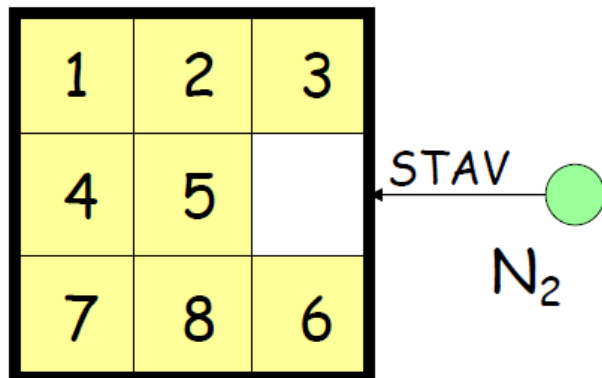
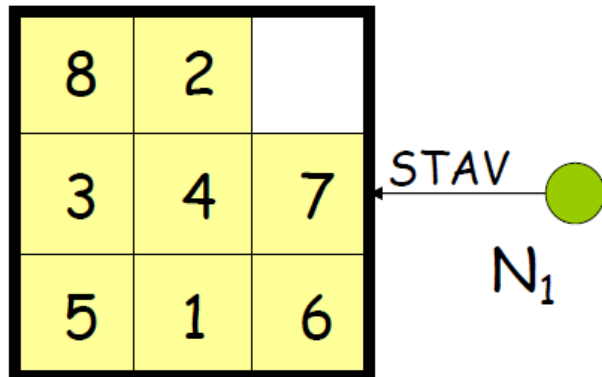
pre slepú stratégiu, N_1 a N_2 sú len dva uzly (s nejakou polohou v strome hľadania)



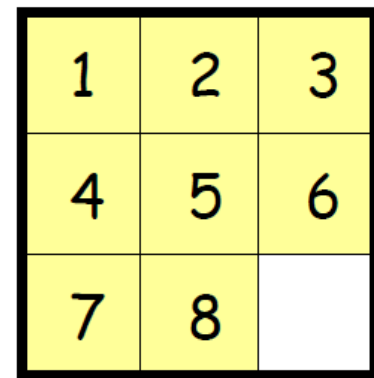
cieľový stav

Príklad

20



pre heuristickú stratégiu, počítajúcu počet kameňov, ktoré nie sú na svojom mieste, N_2 je sľubnejší uzol než N_1



cieľový stav

Poznámka

21

- 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

Slepé stratégie

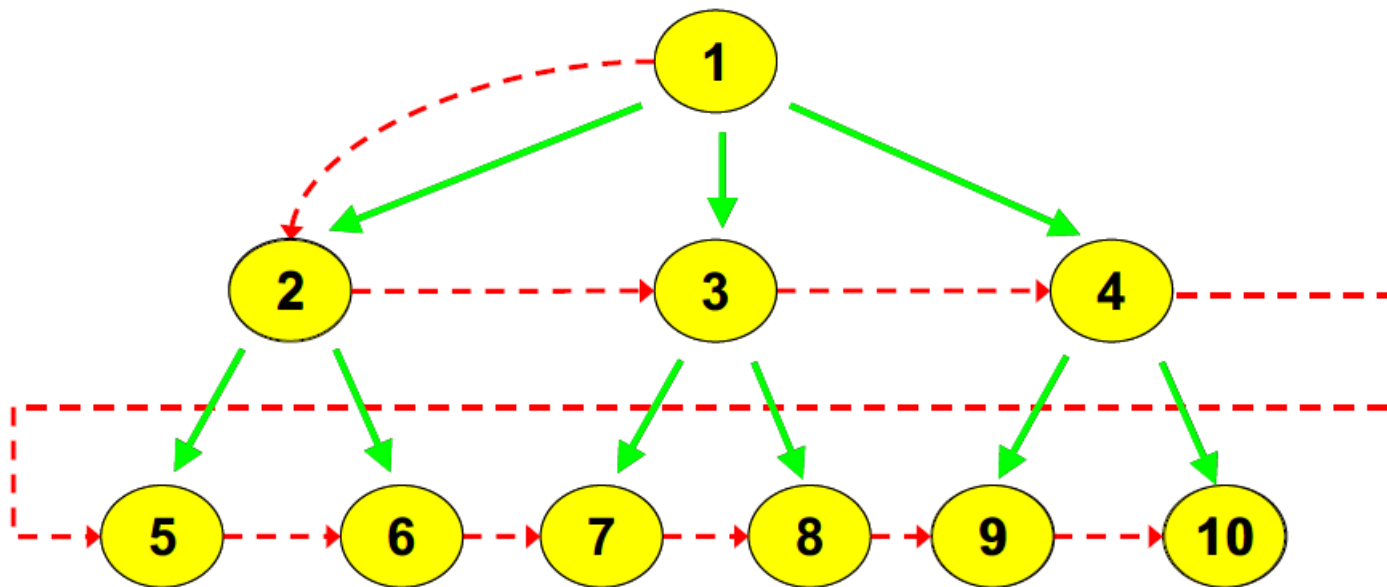
22

- do šírky (Breadth-First Search)
 - obojsmerne
 - do hĺbky (Depth-First Search)
 - obmedzené
 - iteratívne sa prehľbujúce
 - do hĺbky s návratom
 - rovnomerná cena (varianta do šírky)
- cena hrany = 1
- cena hrany
= $c(\text{operátor}) \geq \varepsilon > 0$

Hľadanie do šírky

23

- úplné, prípustné, exponenciálna zložitosť

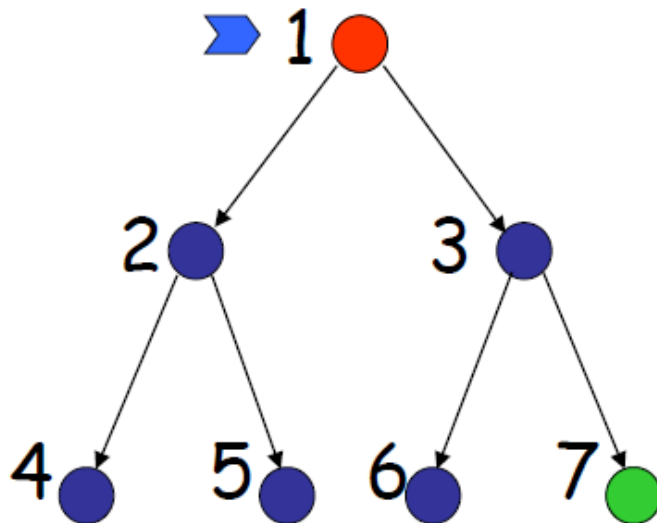


```
function HLADANIE-DO-ŠÍRKY(problém) returns riešenie alebo neúspech  
return VŠEOBECNÉ-HLADANIE(problém, ZARAĎ-NA-KONIEC)
```

Hľadanie do šírky

24

Nové uzly sa pridávajú na koniec OKRAJa

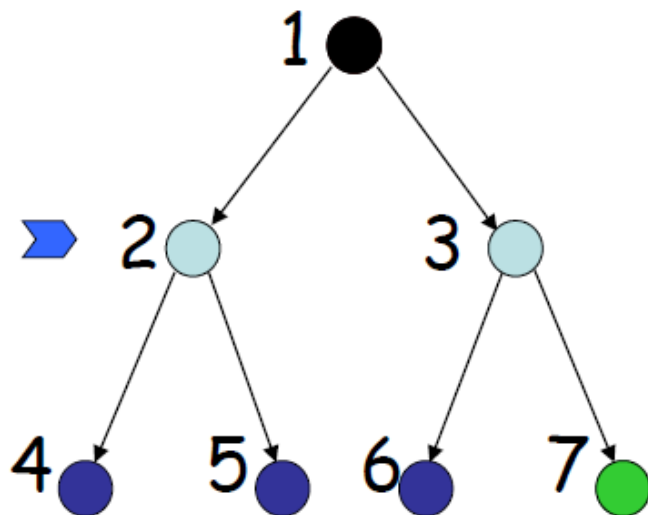


OKRAJ = (1)

Hľadanie do šírky

25

Nové uzly sa pridávajú na koniec OKRAJA

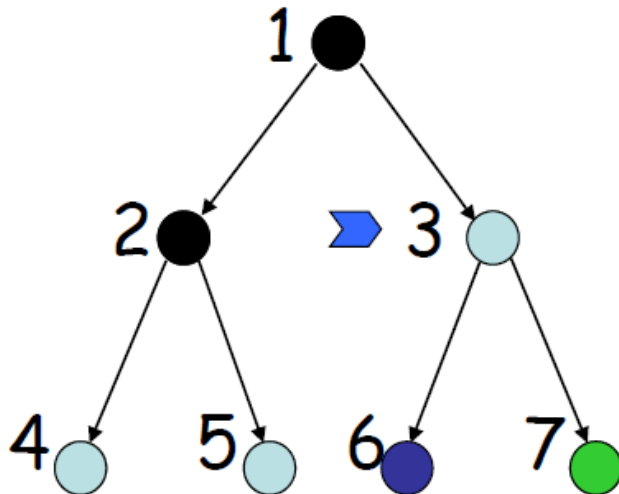


OKRAJ = (2, 3)

Hľadanie do šírky

26

Nové uzly sa pridávajú na koniec OKRAJA

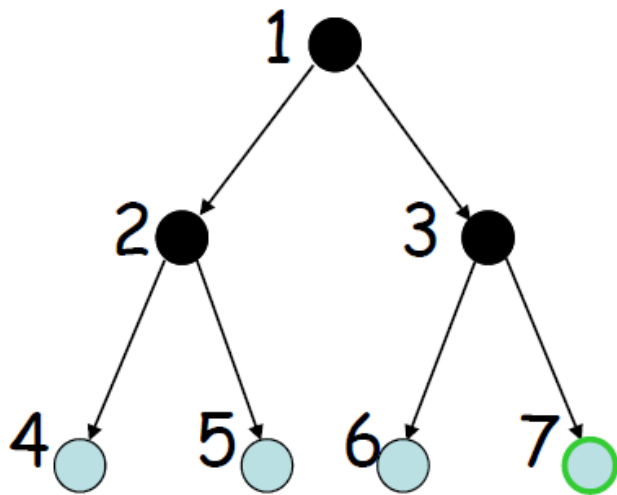


OKRAJ = (3, 4, 5)

Hľadanie do šírky

27

Nové uzly sa pridávajú na koniec OKRAJa



OKRAJ = (4, 5, 6, 7)

Dôležité parametre

28

- 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

Vyhodnotenie

29

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

Vyhodnotenie

30

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

Vyhodnotenie

31

- **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 $O(b^d)$

Časové a pamäťové nároky hľadania do šírky

32

Hĺbka	Počet uzlov	Čas	Pamäť
0	1	0.01 milisekundy	100 slabík
1	35	0.3 milisekundy	3.4 kiloslábík
2	1225	0.01 sekundy	119 kiloslábík
3	42 875	0.4 sekundy	4 megaslabiky
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

Poznámka

33

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)

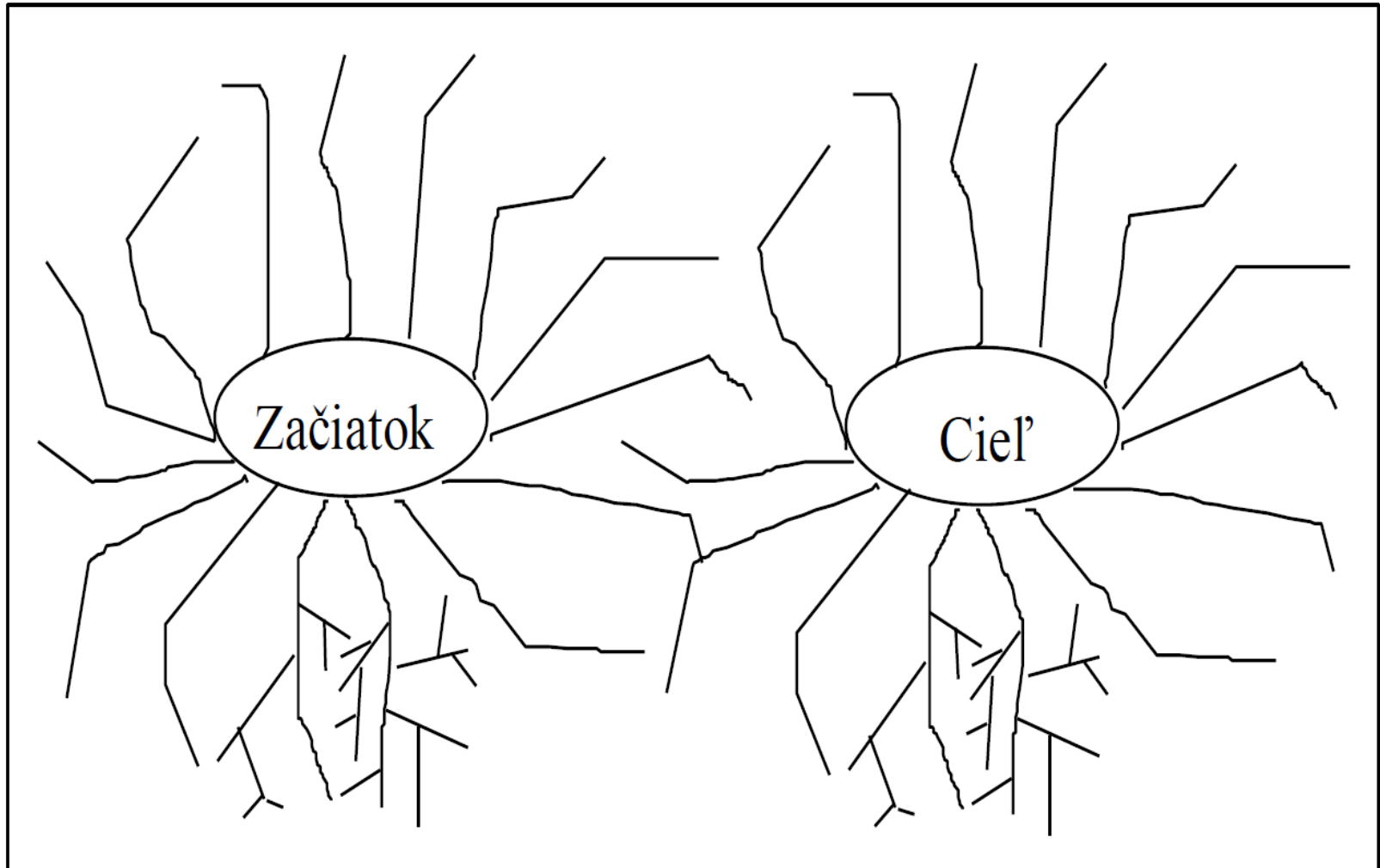
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	



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

Obojsmerné hľadanie

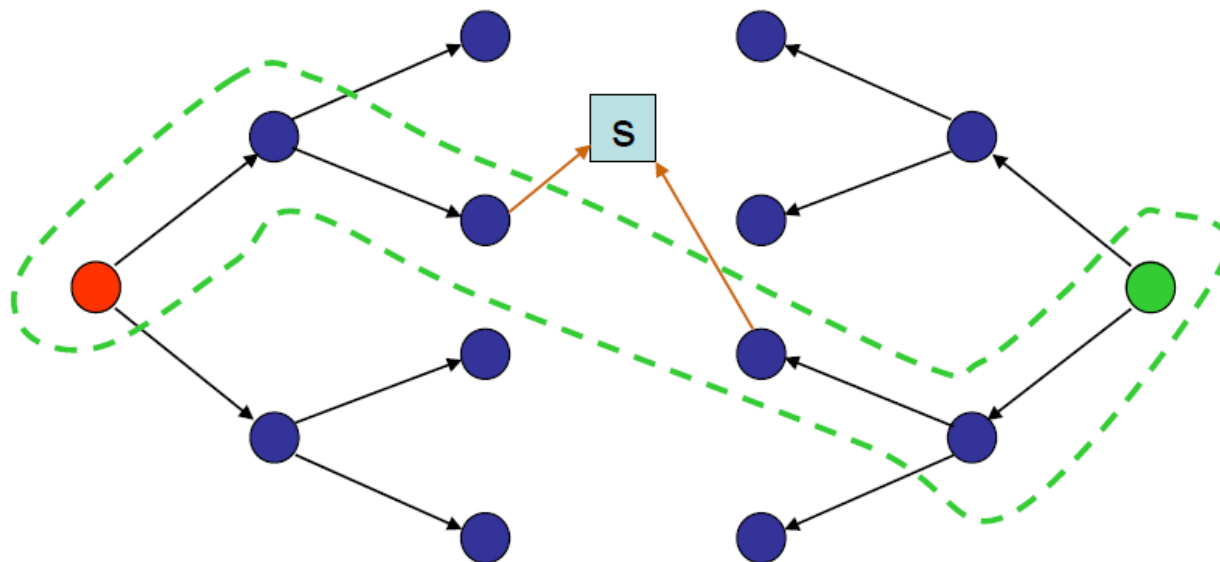
34



Obojsmerná stratégia

35

fronty dvoch okrajov: OKRAJ1 a OKRAJ2



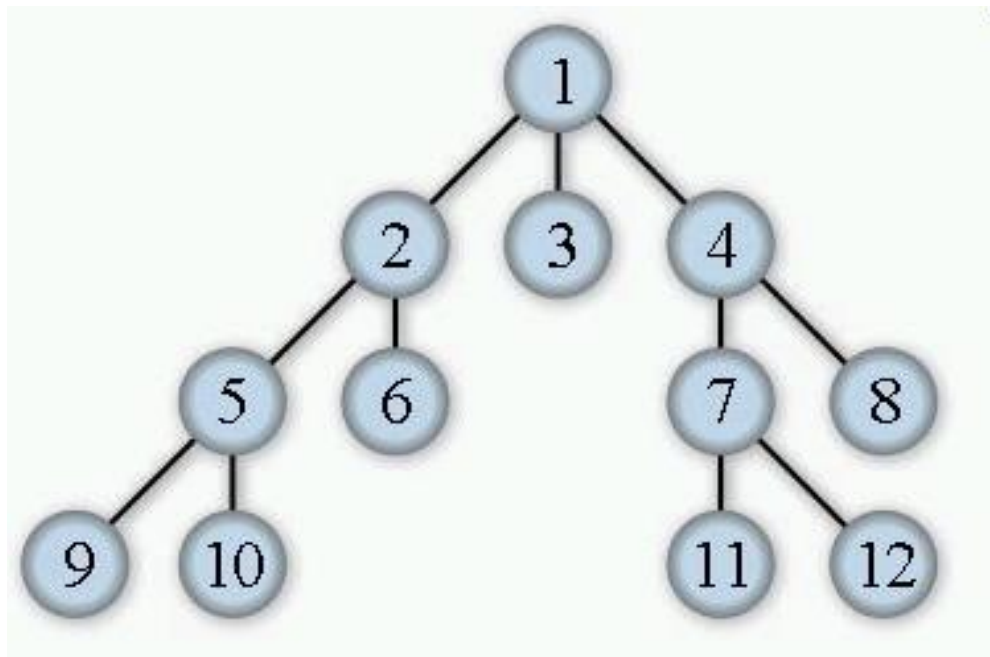
Časová a priestorová zložitosť je $O(b^{d/2}) \ll O(b^d)$
ak oba stromy majú rovnaký vetviaci faktor b

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

Hľadanie do hĺbky

36

□ Depth-first search

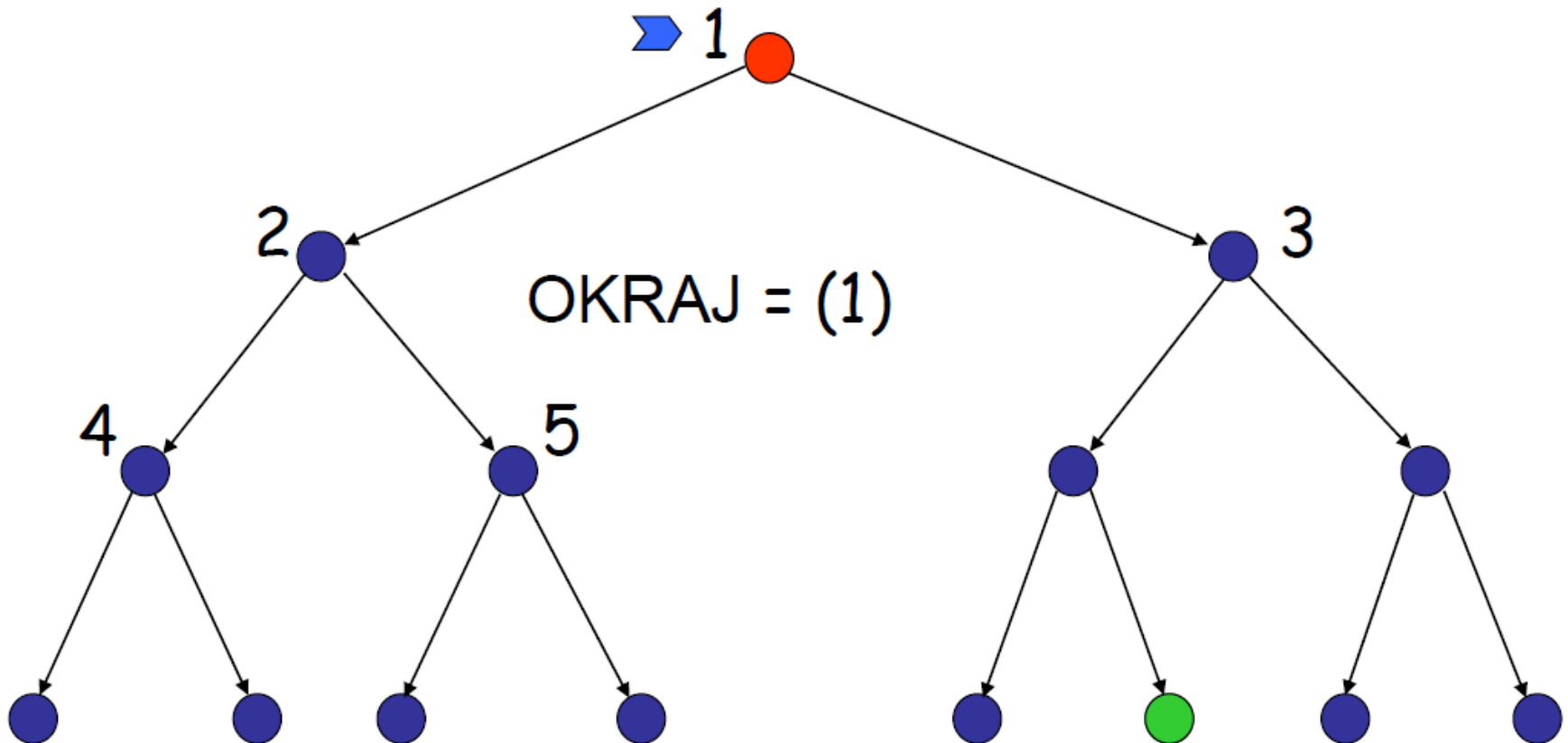


```
function HLADANIE-DO-HLBKY(problém) returns riešenie alebo neúspech  
return VŠEOBECNÉ-HLADANIE(problém, ZARAĎ-NA-ZAČIATOK)
```

Hľadanie do hĺbky

37

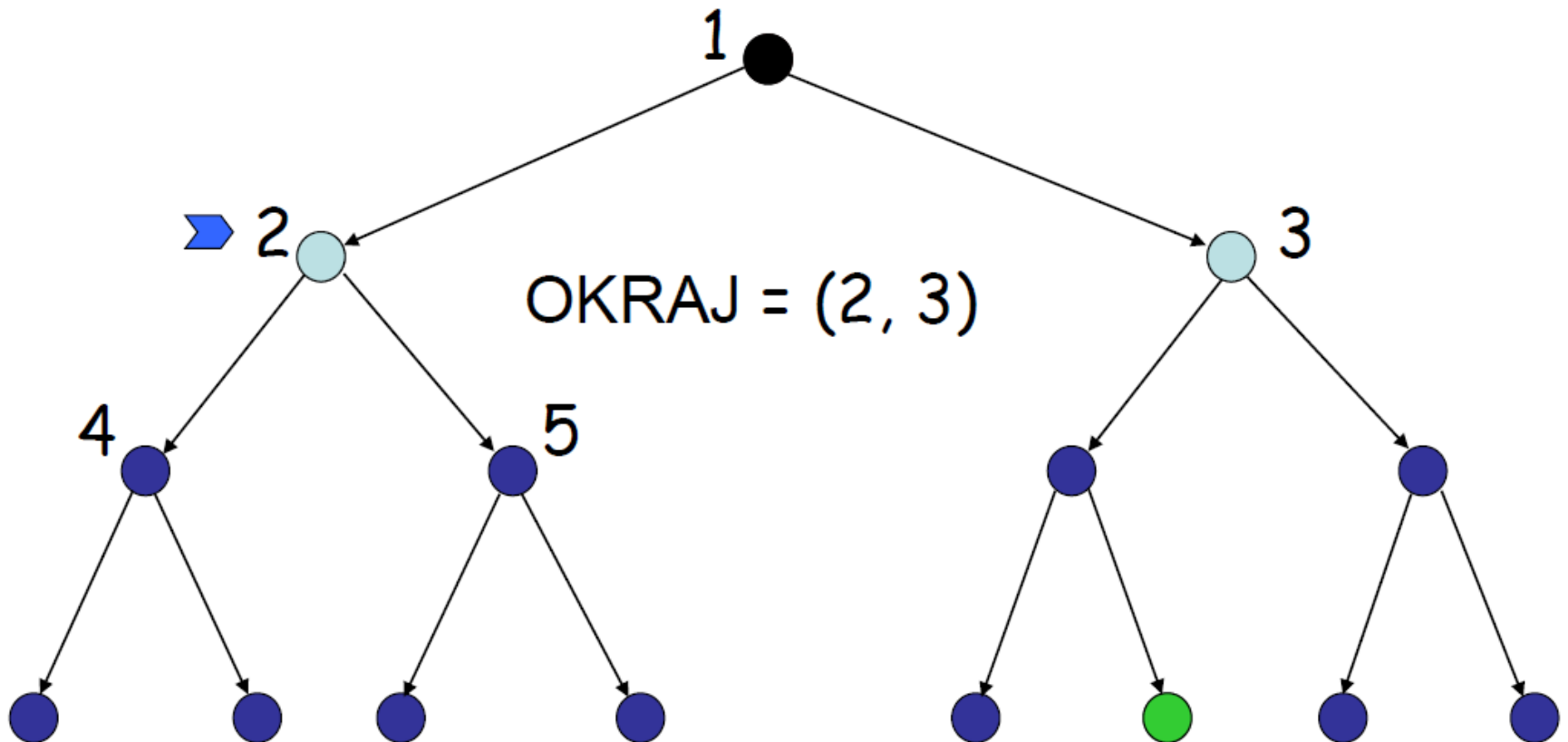
Nové uzly sa vkladajú na začiatok OKRAJA



Hľadanie do hĺbky

38

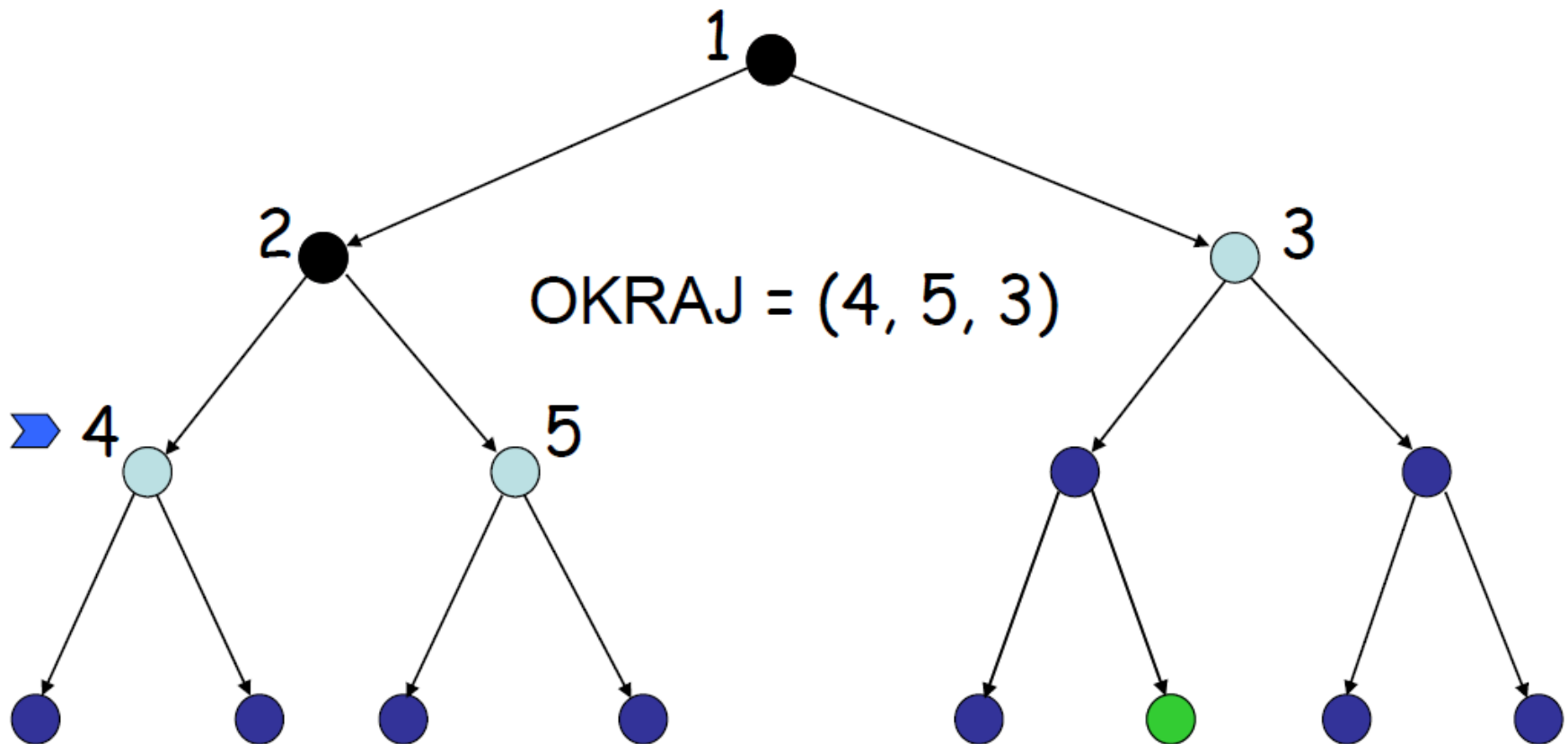
Nové uzly sa vkladajú na začiatok OKRAJA



Hľadanie do hĺbky

39

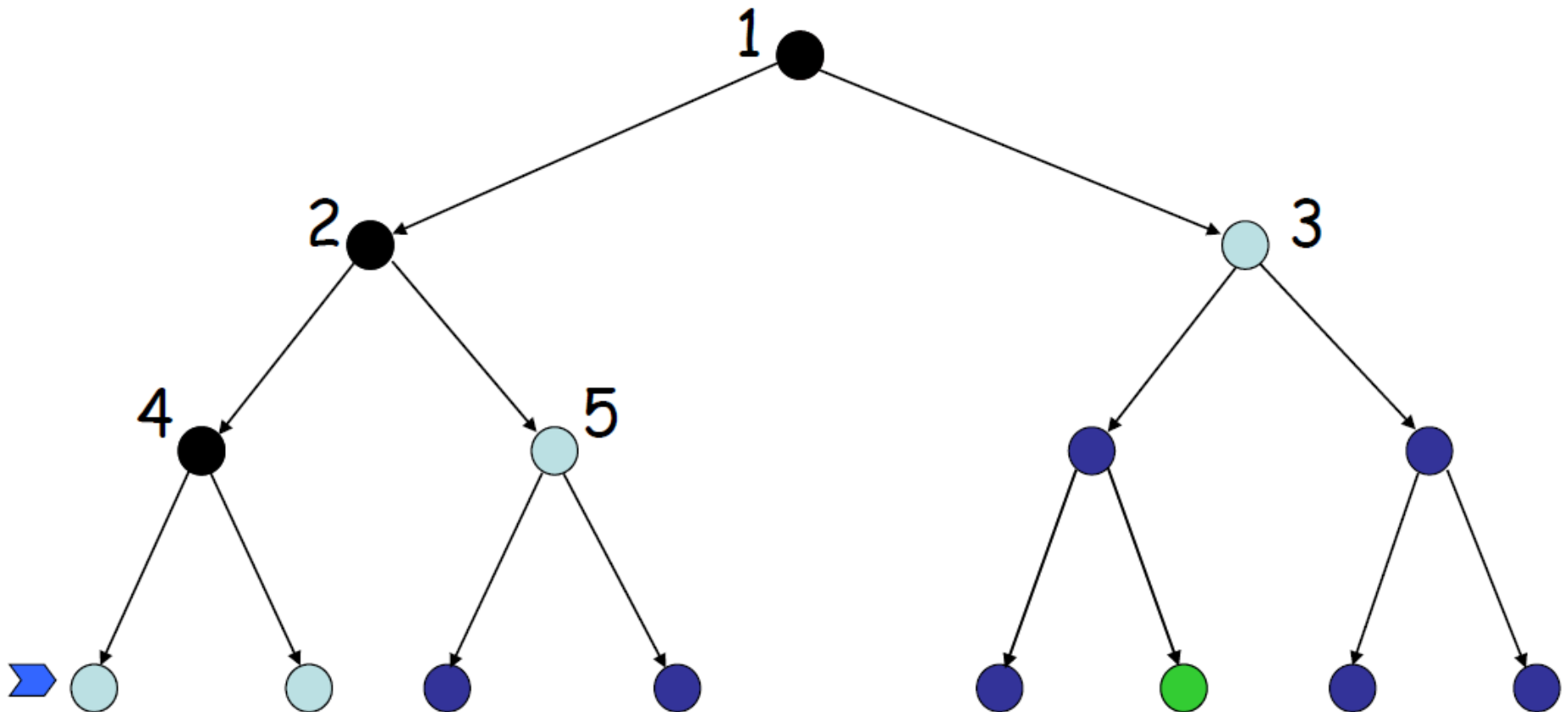
Nové uzly sa vkladajú na začiatok OKRAJA



Hľadanie do hĺbky

40

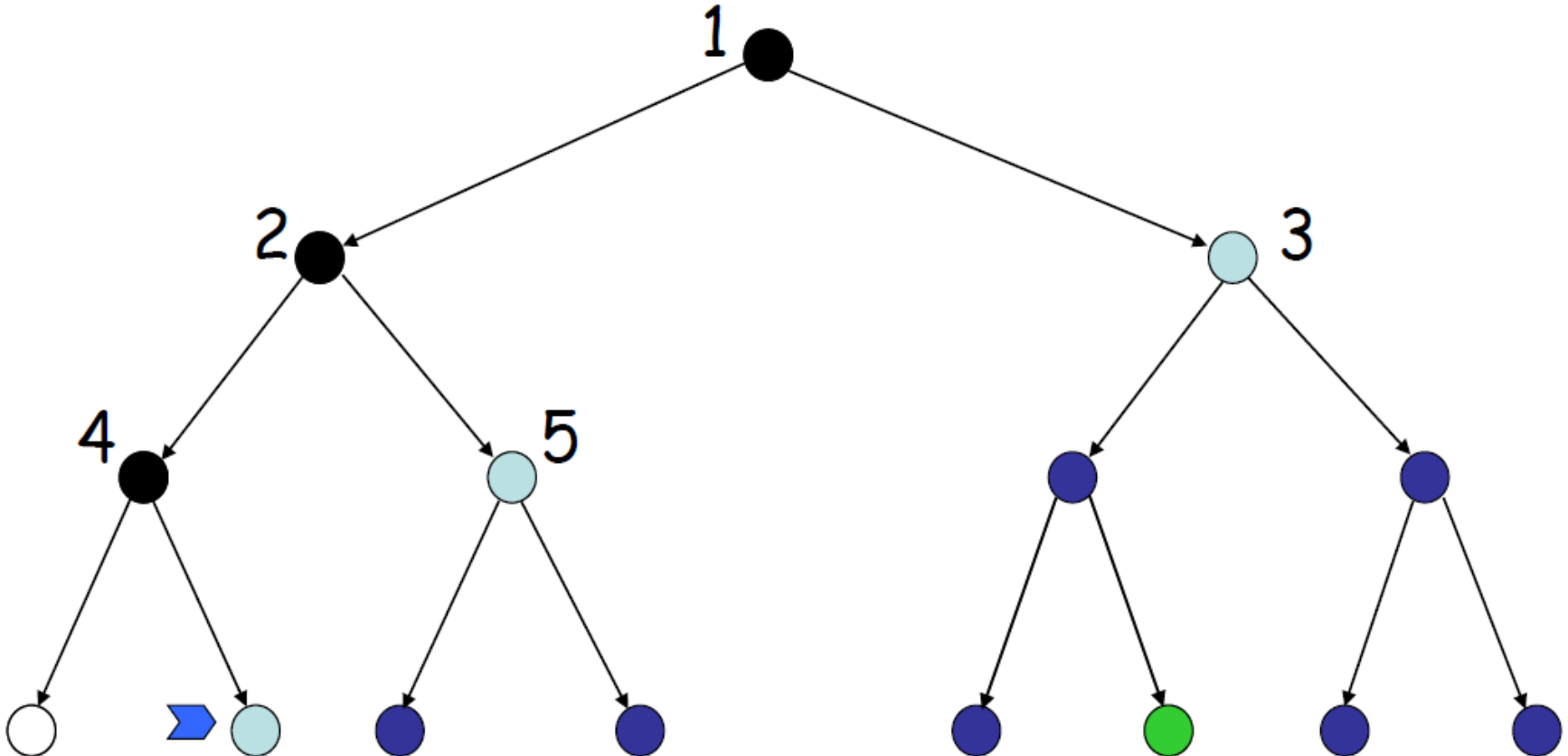
Nové uzly sa vkladajú na začiatok OKRAJa



Hľadanie do hĺbky

41

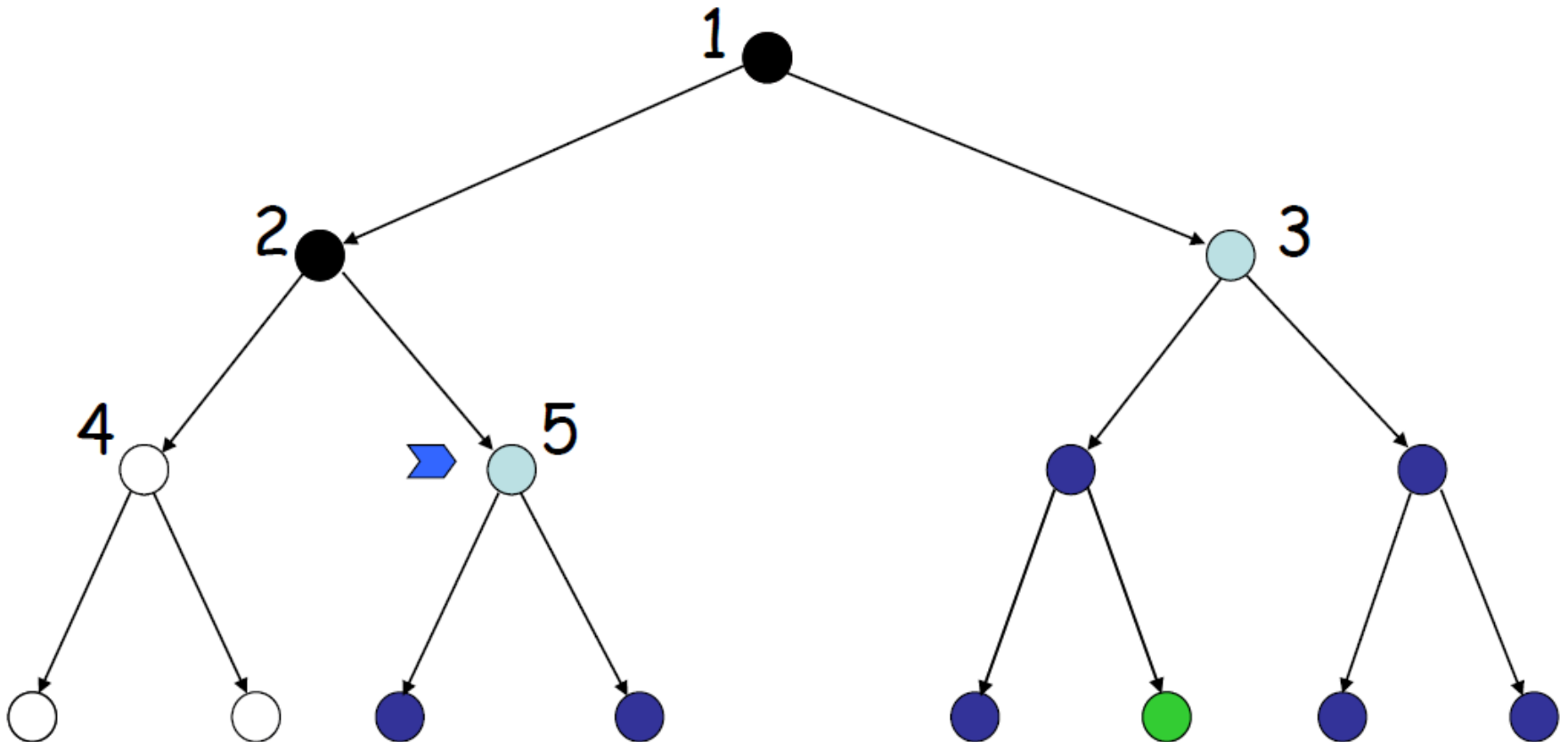
Nové uzly sa vkladajú na začiatok OKRAJa



Hľadanie do hĺbky

42

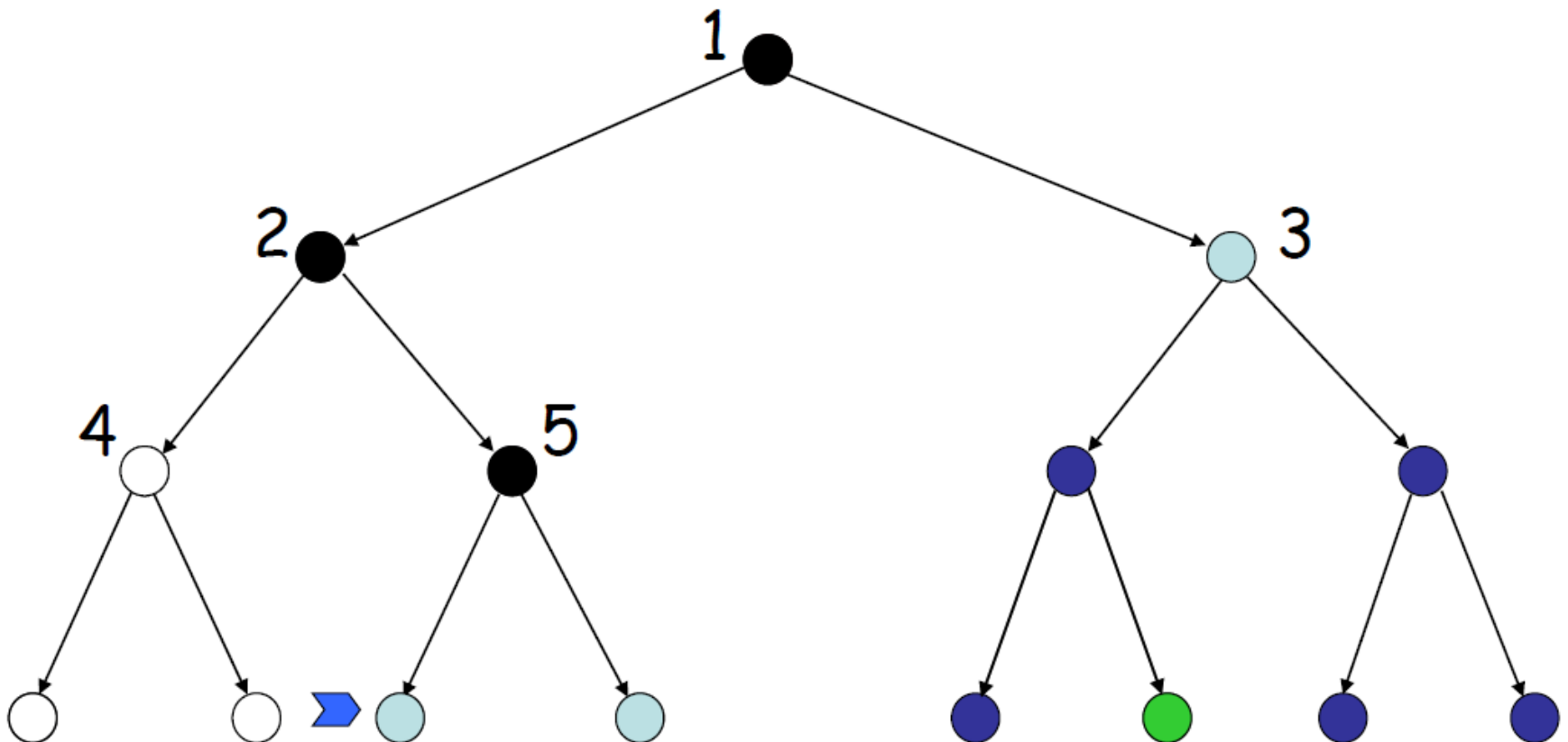
Nové uzly sa vkladajú na začiatok OKRAJA



Hľadanie do hĺbky

43

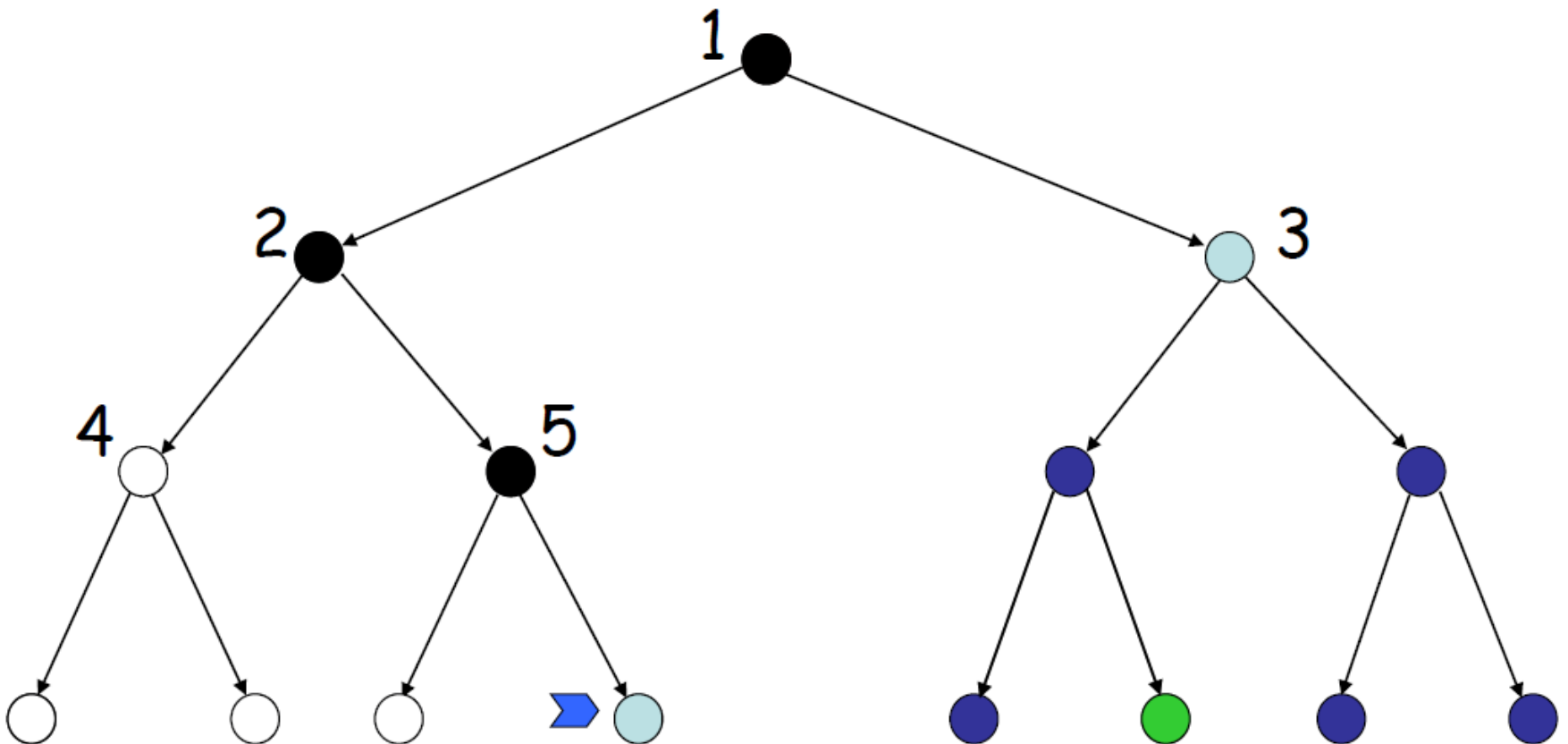
Nové uzly sa vkladajú na začiatok OKRAJa



Hľadanie do hĺbky

44

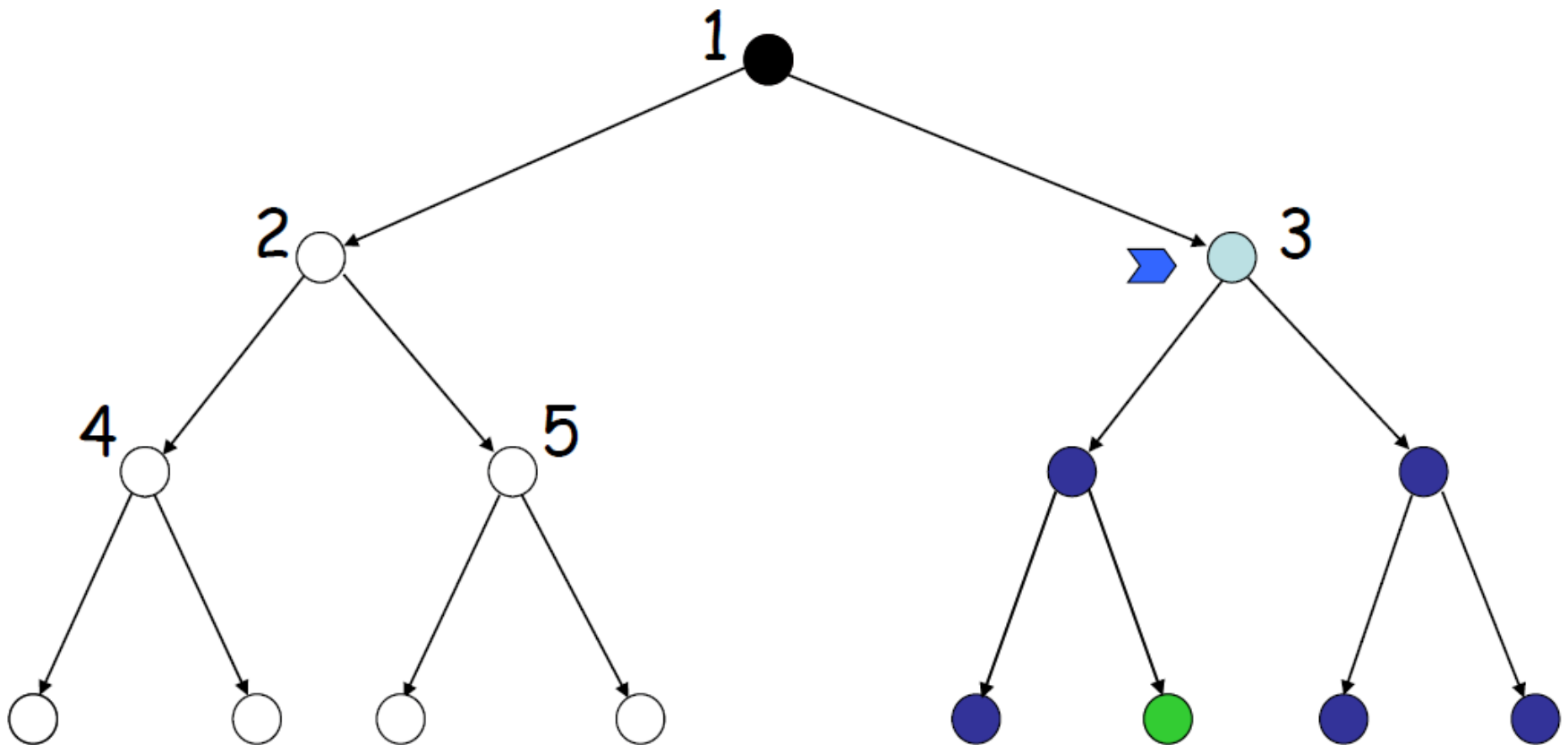
Nové uzly sa vkladajú na začiatok OKRAJA



Hľadanie do hĺbky

45

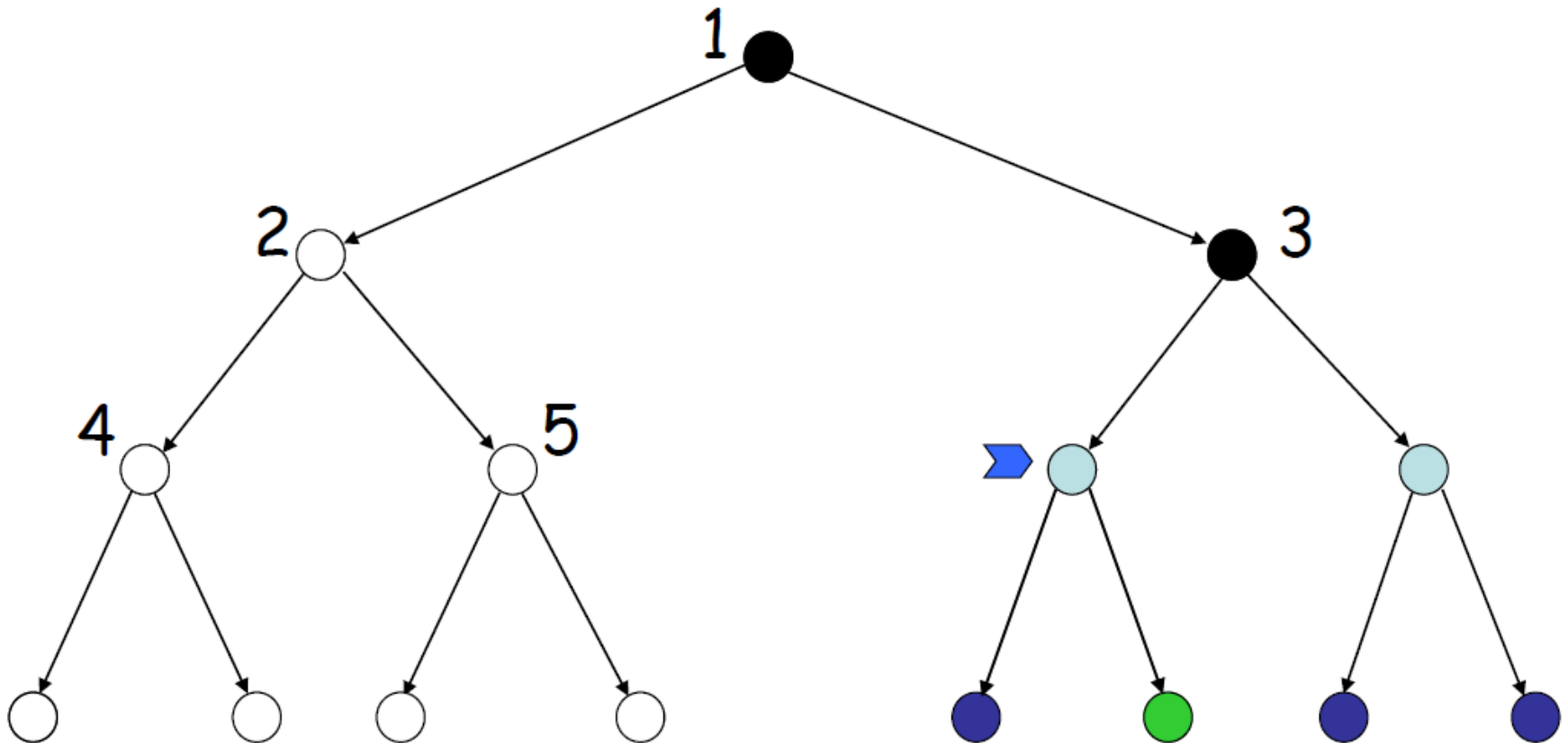
Nové uzly sa vkladajú na začiatok OKRAJa



Hľadanie do hĺbky

46

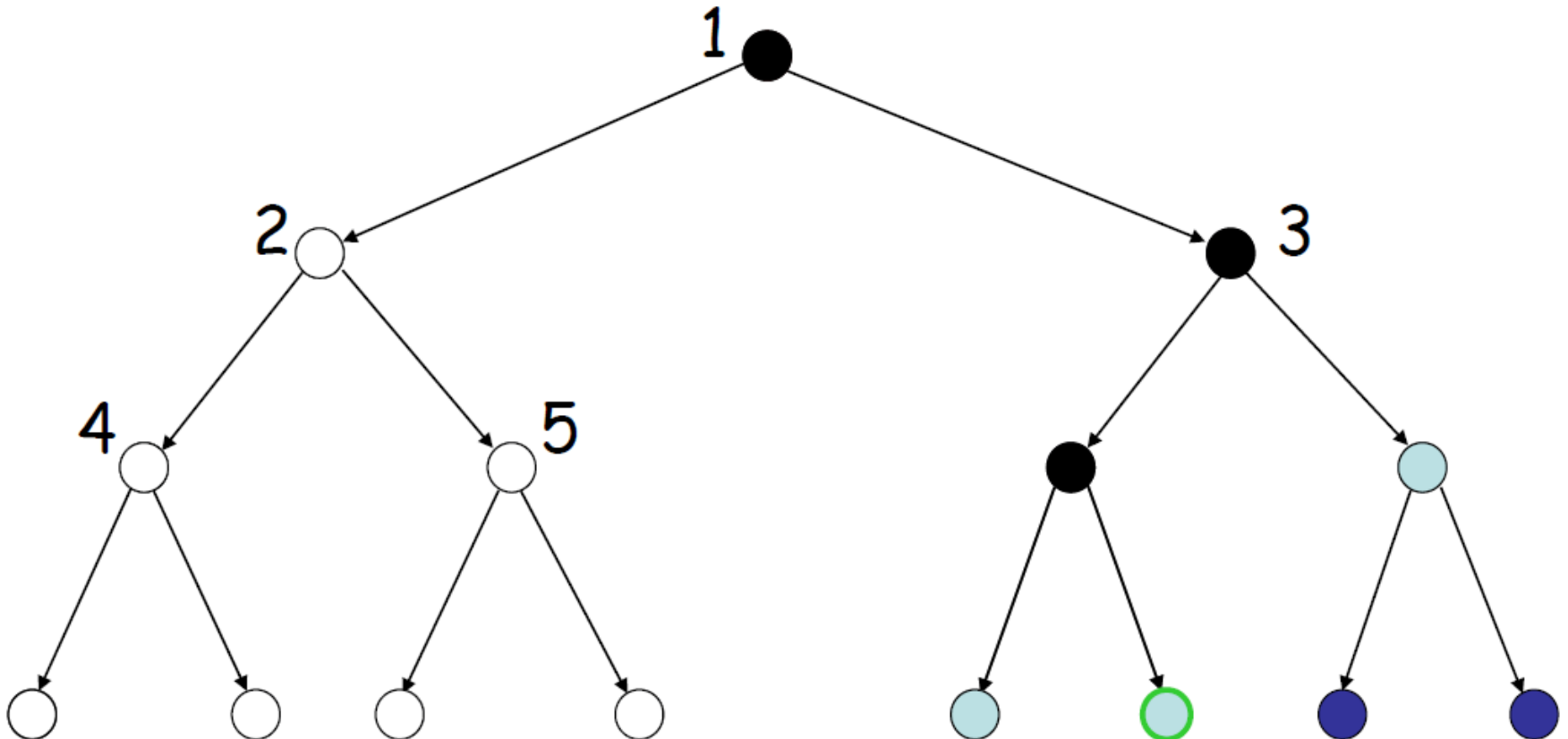
Nové uzly sa vkladajú na začiatok OKRAJA



Hľadanie do hĺbky

47

Nové uzly sa vkladajú na začiatok OKRAJa



Vyhodnotenie

48

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

Vyhodnotenie

49

- **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]

Cyklicky sa prehlbujúce hľadanie

50

```
function CYKLICKY-SA-PREHLBUJÚCE-HĽADANIE(problém)  
returns riešenie alebo neúspech  
  
  for hlbka  $\leftarrow$  0 to  $\infty$  do  
    if OBMEDZENÉ-HĽADANIE(problém, hlbka) je úspešné  
      then return jeho riešenie  
  
  end  
  return neúspech
```

Obmedzené prehľadávanie do hĺbky

51

- Hľadanie do hĺbky s odseknutím v hĺbke k
 - ▣ hĺbka, za ktorou sa uzly nerozvíjajú
- Tri možné prípady
 - ▣ riešenie
 - ▣ zlyhanie – žiadne riešenie
 - ▣ odseknutie hĺbky

Cyklicky sa prehľbujúce hľadanie

52

Poskytuje to najlepšie z hľadania do šírky a do hĺbky

Hlavná idea:

IDS

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

Vykonaj hľadanie do hĺbky s odseknutím v hĺbke k

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

Vyhodnotenie

53

- Cyklicky sa prehľbujúce 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 $O(d)$

Počet generovaných uzlov (hľadanie do šírky a cyklické prehlbovanie)

54

$$d = 5 \text{ a } b = 2$$

v hĺbke	do šírky	cykl. prehľb.
0	1	$1 \times 6 = 6$
1	2	$2 \times 5 = 10$
2	4	$4 \times 4 = 16$
3	8	$8 \times 3 = 24$
4	16	$16 \times 2 = 32$
5	32	$32 \times 1 = 32$
spolu	63	120

$$120/63 \sim 2$$

Počet generovaných uzlov (hľadanie do šírky a cyklické prehlbovanie)

55

$$d = 5 \text{ a } b = 10$$

v hĺbke	do šírky	cykl. prehľb.
0	1	6
1	10	50
2	100	400
3	1,000	3,000
4	10,000	20,000
5	100,000	100,000
spolu	111,111	123,456

$123,456 / 111,111 \sim 1.111$ t.j. len o 11% viac generovaných uzlov

Porovnanie stratégií

56

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

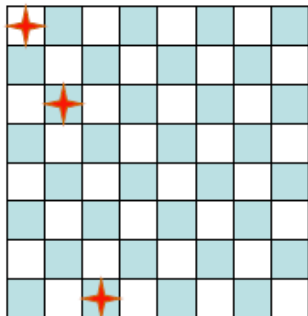
Znovunavšτίvené stavy

57

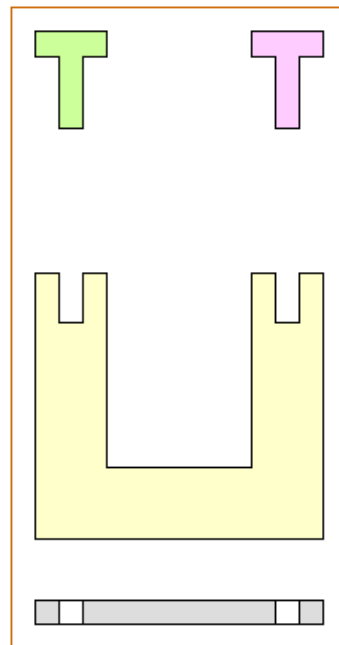
Žiadne

Málo

Veľa

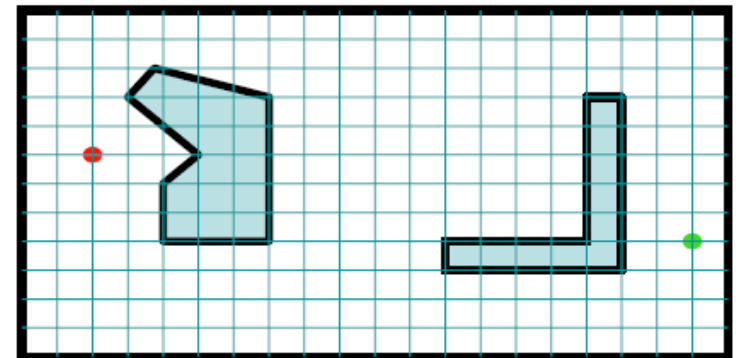


8 dām



plánovanie
montáže

1	2	3
4	5	
7	8	6



8-puzľa a navigácia robota

Znovunavštievané stavy

58

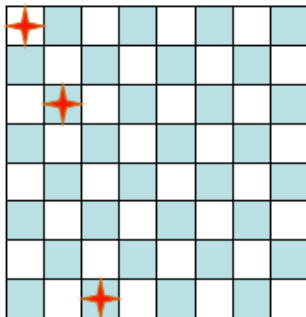
Žiadne

Málo

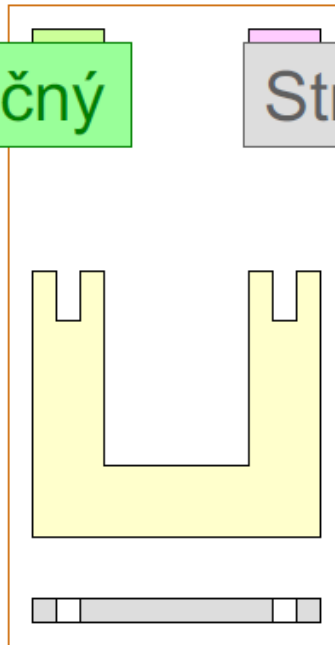
Veľa

Strom je konečný

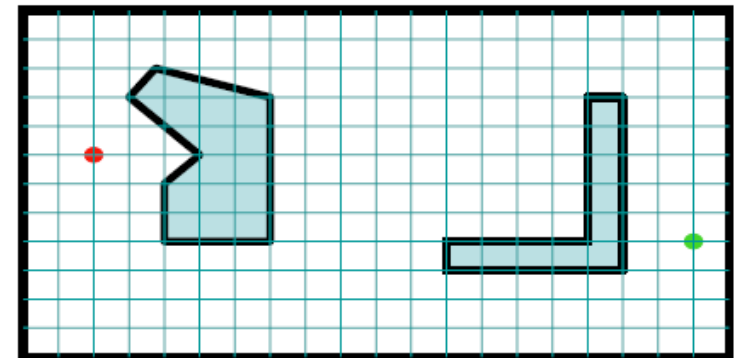
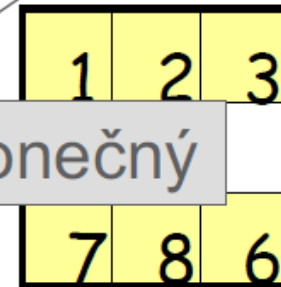
Strom je nekonečný



8 dám



plánovanie
montáže



8-puzľa a navigácia robota

Vyhýbanie sa znovunavštvíveným stavom

59

- 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

Vyhýbanie sa znovunavšτίveným stavom

60

□ 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!

Porovnanie neinformovaných stratégií hľadania

61

Kritérium	Do šírky	Rovnomernej ceny	Do hĺbky	Obmedzené do hĺbky	Cyklicky sa prehĺbujúce	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	$b^{d/2}$
Prípustná?	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)



ĎAKUJEM ZA POZORNOSŤ