



Politechnika Wroclawska

# Rozwiązywanie problemów metodą przeszukiwania

Dariusz Banasiak

Katedra Informatyki Technicznej W4/K9

Politechnika Wroclawska

Jedną z ważniejszych metod sztucznej inteligencji jest szukanie (ang. searching).

Metodą przeszukiwania można rozwiązywać następujące typy zadań:

- przesuwanki np. kostka Rubika, puzzle
- labirynty, poszukiwanie optymalnej drogi
- problemy układania klocków np. wieża z Hanoi
- zagadki logiczne np. misjonarze i kanibale
- gry planszowe
- inne

Aby zastosować metodę przeszukiwania należy w odpowiedni sposób określić przestrzeń poszukiwań.

W tym celu należy zdefiniować trzy elementy:

- bazę danych (fakty, stany, możliwości, opis sytuacji)
- możliwe operacje, które zmieniają stan bazy danych
- strategię kontrolną (start, koniec i kolejność operacji)

Przestrzeń stanów definiowana jest jako uporządkowana czwórka  $[N, A, S, G]$ , gdzie:

- $N$  jest zbiorem wierzchołków odpowiadających stanom w procesie rozwiązywania problemu
- $A$  jest zbiorem krawędzi odpowiadających krokom (operacjom) w procesie rozwiązywania problemu
- $S$  jest niepustym podzbiorem  $N$ , zawierającym stany początkowe problemu
- $G$  jest niepustym podzbiorem  $N$ , zawierającym stany docelowe (końcowe) problemu

## Reprezentacja problemu w przestrzeni stanów

Bardzo wygodnym sposobem reprezentacji przestrzeni stanów są grafy lub drzewa (czasami stosowane są specjalne typy grafów np. AND/OR). Ścieżką rozwiązania danego problemu jest wówczas ścieżka wiodąca przez ten graf z wierzchołka S do wierzchołka G.

W przypadku niektórych problemów z każdym przejściem między stanami związany jest określony koszt. W procesie szukania należy dążyć wówczas do znalezienia ścieżki o minimalnym całkowitym koszcie.

## Reprezentacja problemu w przestrzeni stanów

Graf (drzewo) opisujące wszystkie możliwe sytuacje wyznacza przestrzeń szukania. W zależności od złożoności problemu przestrzeń szukania może mieć różny rozmiar (może być również nieskończona).

- puzzle (3x3)       $\longrightarrow$        $9!/2 = 181440$  wierzchołków
- warcaby             $\longrightarrow$        $10^{40}$  stanów końcowych
- szachy               $\longrightarrow$        $10^{120}$  stanów końcowych

### Problem 1 – Lis, gęś i ziarno

Nad rzeką znajduje się farmer z lisem, gęsią i ziarnem, które musi przewieźć na drugi brzeg. Zadanie posiada następujące ograniczenia:

- w danej chwili można przewieźć łódką tylko jedną rzecz
- bez opieki nie mogą zostać lis z gęsią oraz gęś z ziarnem

## Reprezentacja problemu w przestrzeni stanów

**F L G Z**

**Opis stanu:**  $(x_1, x_2, x_3, x_4)$ , gdzie  $x_i = \{L, P\}$

**Stan początkowy:** (L, L, L, L)

**Stan końcowy:** (P, P, P, P)

### **Dopuszczalne operacje:**

1. Farmer płynie sam
2. Farmer przewozi lisa
3. Farmer przewozi gęś
4. Farmer przewozi ziarno



## Reprezentacja problemu w przestrzeni stanów

Liczba wszystkich stanów:  $2^4 = 16$

Liczba stanów akceptowalnych: 10

Liczba stanów nieakceptowalnych: 6

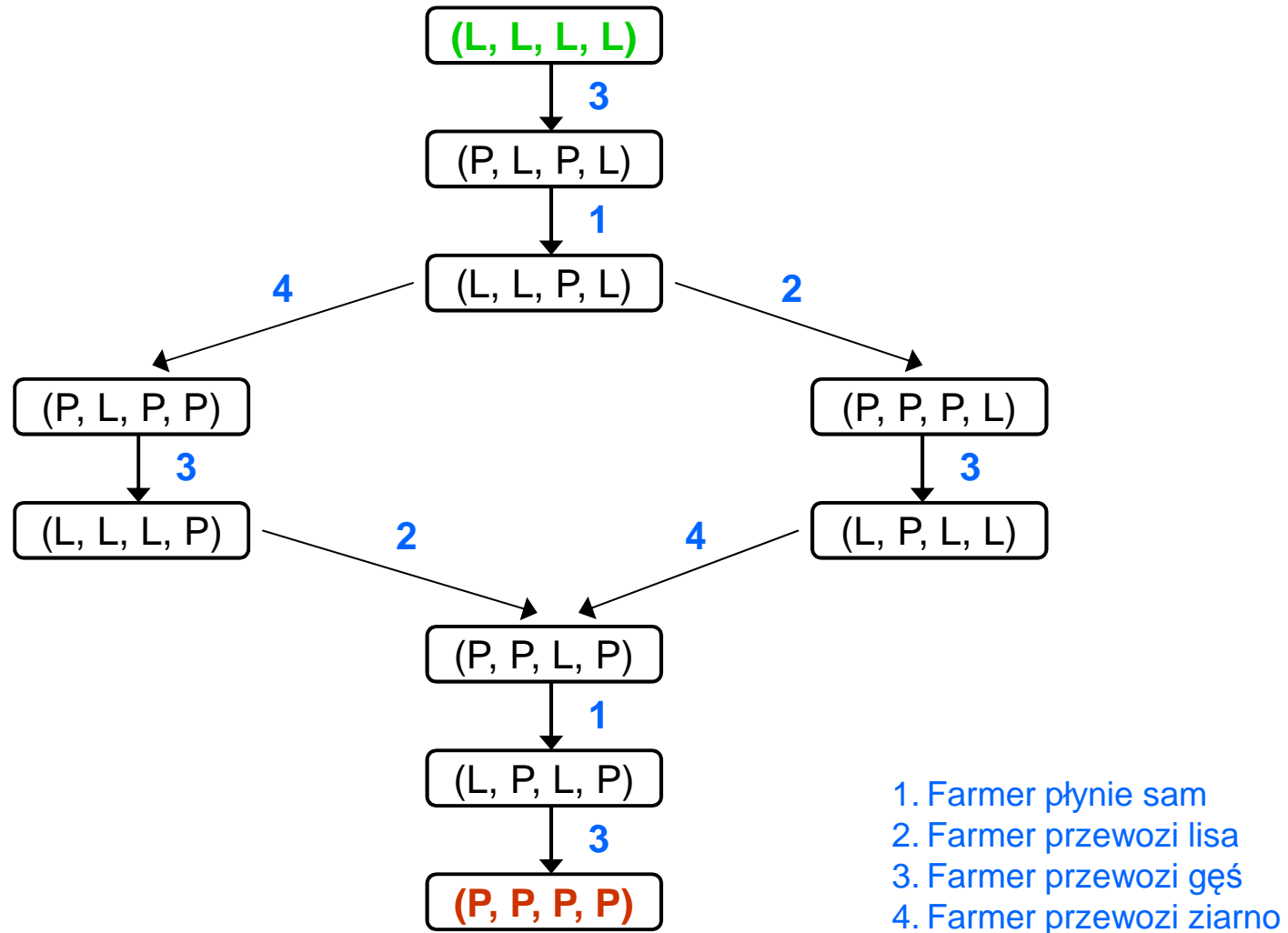
Przykłady stanów nieakceptowalnych:

$(P, L, L, P), (L, P, P, L)$  – lis i gęś razem „bez opieki”

$(P, P, L, L), (L, L, P, P)$  – gęś i ziarno razem „bez opieki”

Na następnym rysunku przedstawiono rozwiązanie problemu, które zawiera jedynie stany akceptowalne. Pominięto stany „ślepe”, które nie prowadzą do rozwiązania.

## Reprezentacja problemu w przestrzeni stanów



### Problem 2 – Napełnianie naczyń

Posiadamy dwa puste dzbanki:

- o pojemności 4 l (dzbanek 1)
- o pojemności 3 l (dzbanek 2)

Dzbanek 1 (o pojemności 4 l) należy napełnić do połowy.

**Opis stanu:**  $(x, y)$    $x$  – ilość wody w dzbanku 1  
 $y$  – ilość wody w dzbanku 2

**Stan początkowy:**  $(0, 0)$

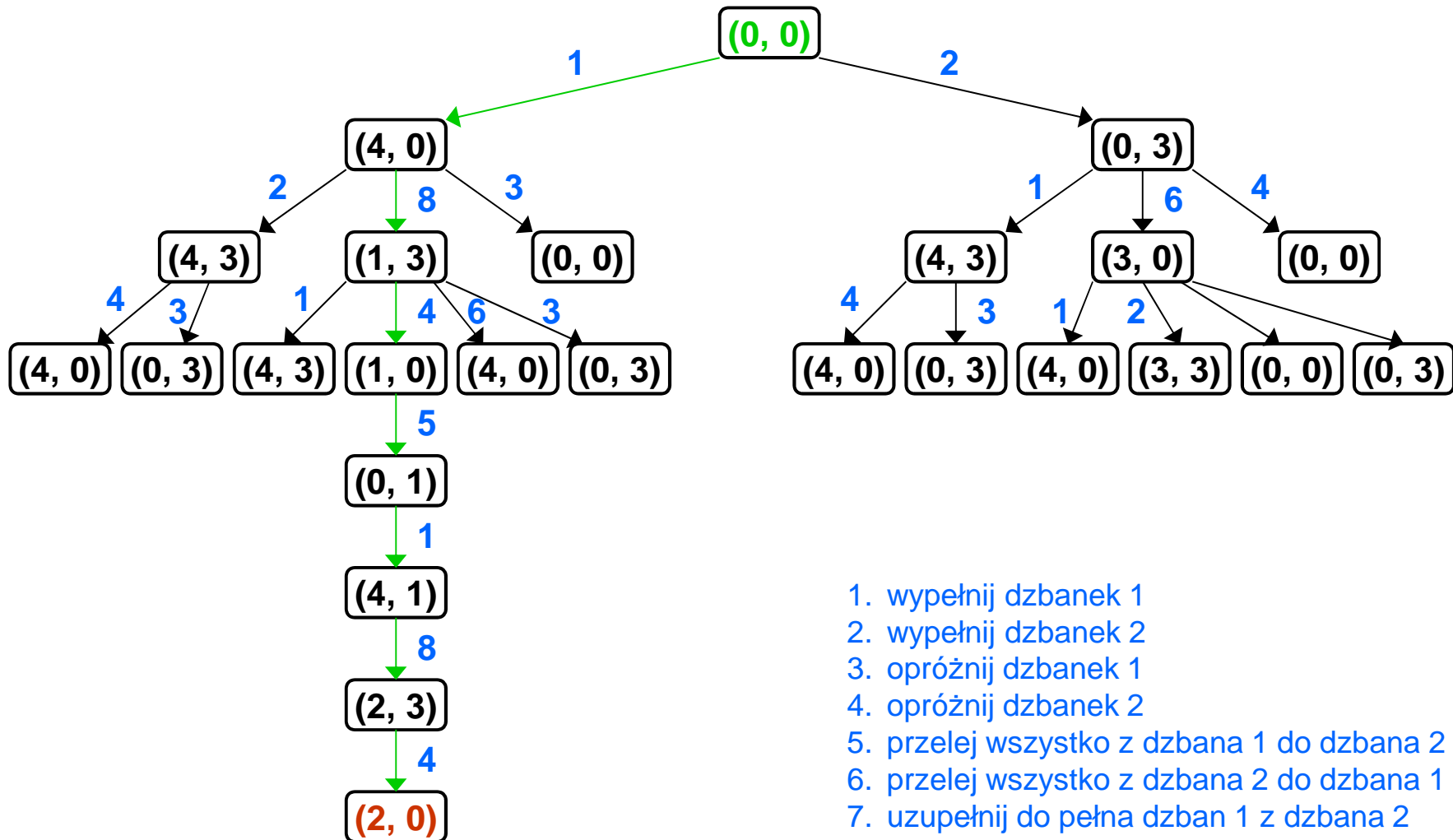
**Stan końcowy:**  $(2, 0)$

### Dopuszczalne operacje:

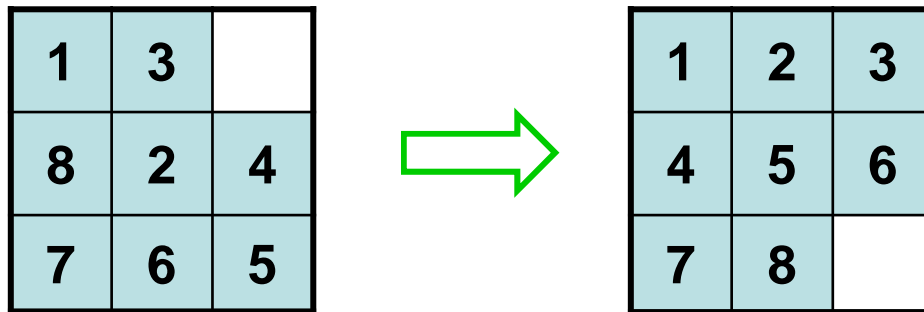
1. wypełnij dzbanek 1  $((x, y) \rightarrow (4, y), \text{warunek: } x < 4)$
2. wypełnij dzbanek 2  $((x, y) \rightarrow (x, 3), \text{warunek: } y < 3)$
3. opróżnij dzbanek 1  $((x, y) \rightarrow (0, y), \text{warunek: } x > 0)$
4. opróżnij dzbanek 2  $((x, y) \rightarrow (x, 0), \text{warunek: } y > 0)$
5. przelej wszystko z dzbana 1 do dzbana 2  $((x, y) \rightarrow (0, x+y), \text{war.: } x > 0, x+y \leq 3)$
6. przelej wszystko z dzbana 2 do dzbana 1  $((x, y) \rightarrow (x+y, 0), \text{war.: } y > 0, x+y \leq 4)$
7. uzupełnij do pełna dzban 1 z dzbana 2  $((x, y) \rightarrow (4, y-(4-x)), \text{war.: } y > 0, x+y \geq 4)$
8. uzupełnij do pełna dzban 2 z dzbana 1  $((x, y) \rightarrow (x-(3-y), 3), \text{war.: } x > 0, x+y \geq 3)$
9. odlej pewną ilość wody z dzbana 1  $((x, y) \rightarrow (x-d, y), \text{war.: } x > 0)$
10. odlej pewną ilość wody z dzbana 2  $((x, y) \rightarrow (x, y-d), \text{war.: } y > 0)$
11. ...

Można definiować kolejne operacje. Do rozwiązania problemu wystarczą operacje 1-8.

# Reprezentacja problemu w przestrzeni stanów



## Problem 3 – Puzzle 3x3 (tzw. ósemka)

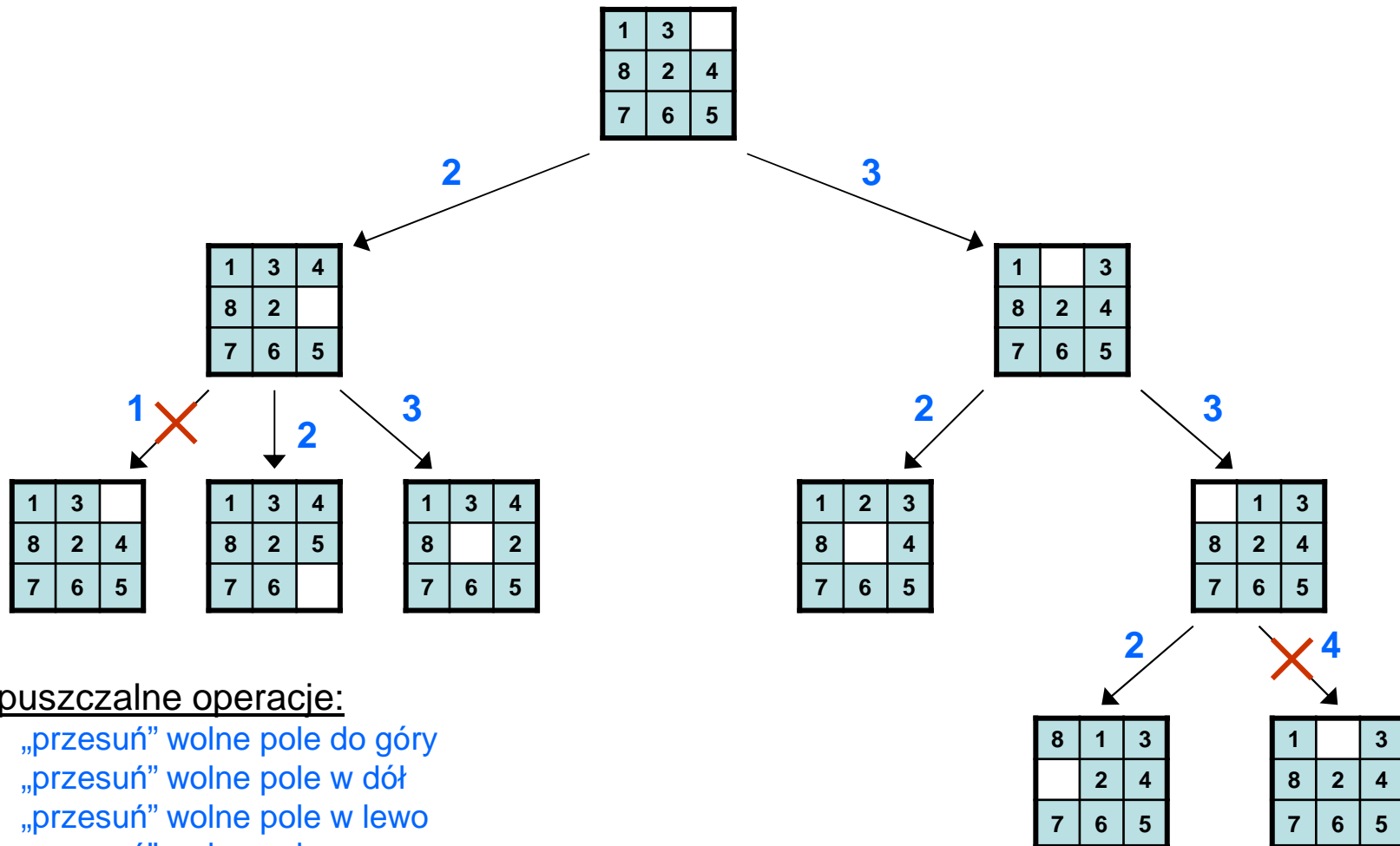


**Opis stanu:**  $(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)$ ,  
gdzie  $x_i = \{0, 1, \dots, 8\}$

**Stan początkowy:**  $(?, ?, ?, ?, ?, ?, ?, ?, ?)$

**Stan końcowy:**  $(1, 2, 3, 4, 5, 6, 7, 8, 0)$

# Reprezentacja problemu w przestrzeni stanów



W celu uzyskania rozwiązania danego problemu metodą przeszukiwania należy:

- określić reprezentację problemu w przestrzeni stanu (wyróżnienie poszczególnych stanów, sposób ich kodowania),
- określić dopuszczalne operatory, które pozwalają dla dowolnego stanu określić stany następne,
- zdefiniować stan początkowy i stan (lub stany) docelowe,
- określić strategię przeszukiwania, czyli metodę wyboru operatora spośród wszystkich operatorów możliwych.



Przestrzeń poszukiwań może być reprezentowana w postaci grafu lub drzewa. Rozwiązanie problemu polega wówczas na szukaniu w grafie (drzewie) ścieżki od wierzchołka początkowego do jednego z wierzchołków docelowych:

St := [stan\_początkowy]

**WHILE** ( St <> stan\_docelowy) **DO**

**BEGIN**

Op := [wybrany operator dla stanu St]

St := [wynik zastosowania Op do stanu St]

**END**

Metodę wyboru operatora w powyższym algorytmie nazywa się strategią poszukiwań. Wybierając odpowiednią strategię należy uwzględnić następujące czynniki:

- ograniczenie rozmiaru drzewa poszukiwań (np. eliminacja operatorów nie prowadzących do celu)
- znalezienie optymalnego rozwiązania w przypadku wielu rozwiązań (np. znalezienie najkrótszej ścieżki od wierzchołka początkowego do docelowego)
- znalezienie rozwiązania w najkrótszym czasie
- zakończenie algorytmu w przypadku braku rozwiązania (ograniczenie głębokości poszukiwań)

Podstawowym algorytmem przeszukiwania przestrzeni stanów jest systematyczne badanie grafu (tzw. przegląd zupełny). Celem takiego algorytmu jest sprawdzenie wszystkich wierzchołków grafu począwszy od wierzchołka startowego.

**Rozwiązanie takie jest często nieefektywne!!!**

Stosowane strategie przeszukiwania możemy podzielić na dwie grupy:

- metody szukania „na ślepo” (klasyczne)
- metody heurystyczne

### Metody szukania „na ślepo”

W większości zadań praktycznych rozmiary drzew są tak duże, że nie jest możliwe ich zapamiętanie w całości. Dlatego często stosuje się metodę generowania tych drzew „na bieżąco”.

W zależności od kolejności przeszukiwanych wierzchołków w drzewie rozróżnia się następujące strategie:

- Monte Carlo (metoda Muzeum Brytyjskiego)
- szukanie „w głąb” (DFS - ang. depth-first search)
- szukanie „wszerz” (BFS - ang. breadth-first search)
- szukanie z nawracaniem (BT - ang. backtracking)
- inne

### Szukanie „w głąb” (DFS)

Szukanie „w głąb” jest podstawową metodą przeszukiwania grafów. Przeszukiwanie prowadzone jest zawsze od ostatnio sprawdzanego wierzchołka, który ma jeszcze niewykorzystane operatory.

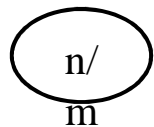
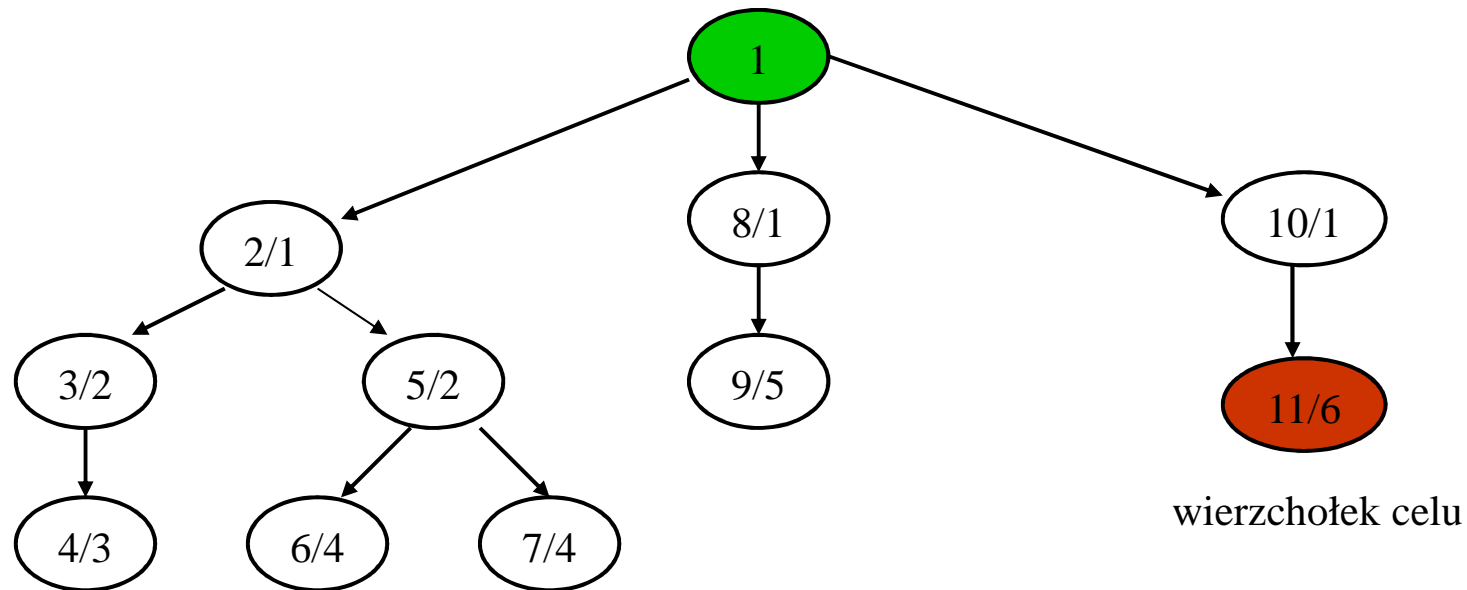
W algorytmie wykorzystuje się listę OPEN, która zawiera wierzchołki drzewa, które nie zostały jeszcze rozpatrzone. Lista OPEN zorganizowana jest w postaci stosu (ostatnio umieszczony element pobierany jest jako pierwszy).

**Metoda może być nieskuteczna dla dużych grafów (np. dla grafów o nieskończonej głębokości)!!!**

### Zapis algorytmu:

1. wstaw wierzchołek początkowy na listę OPEN
2. IF (lista OPEN jest pusta) THEN STOP (brak rozwiązania)
3. St := FIRST (OPEN)
4. IF (St jest wierzchołkiem docelowym) THEN STOP (sukces)
5. usuń St z listy OPEN
6. utwórz wszystkie wierzchołki potomne od St i wstaw je na początek listy OPEN,
7. GOTO 2

## Działanie algorytmu:



$n$  – liczba określająca kolejność analizy wierzchołków,  
 $m$  – liczba określająca kolejność dodawania nowych wierzchołków do listy OPEN.

### Szukanie „wszerz” (BFS)

Szukanie „wszerz” polega na rozwijaniu na każdym poziomie wszystkich wierzchołków następnej generacji (określenie wszystkich potomków dla wszystkich wierzchołków z danego poziomu i ich sprawdzenie).

Metoda ta, kolejno badając wierzchołki grafu o jednakowej głębokości, przyznaje wyższy priorytet wierzchołkom o mniejszej głębokości. Jeżeli rozwiązanie położone jest niezbyt „głęboko” procedura szukania „wszerz” szybciej prowadzi do jego znalezienia.



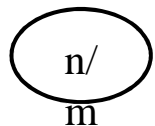
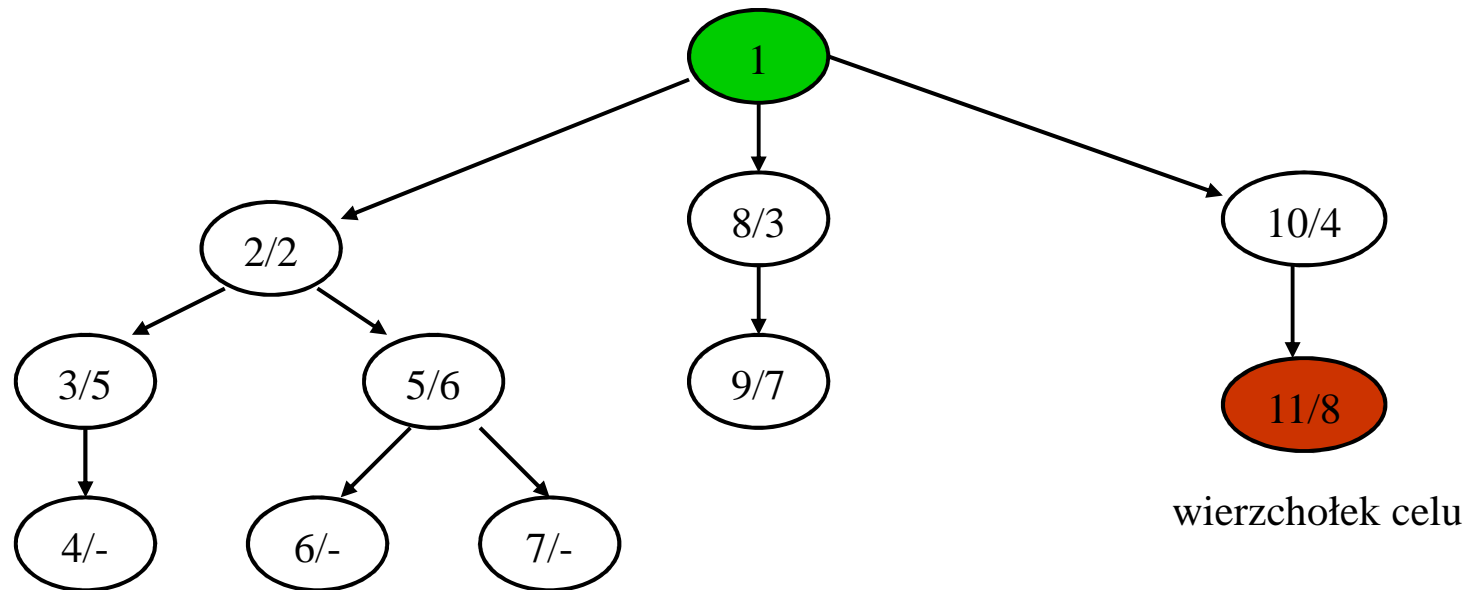
Algorytm szukania „wszerz” jest podobny do algorytmu przedstawionego dla strategii szukania „w głąb”. Podstawowa różnica polega na sposobie organizacji listy OPEN. Podczas szukania „wszerz” lista OPEN zorganizowana jest w postaci **kolejki**.

Strategia ta wyznacza wierzchołek celu o najmniejszej głębokości.

Wadą strategii są duże wymagania pamięciowe (należy przechowywać całą dotychczas przebadaną część grafu).

# Algoritmy przeszukiwania przestrzeni stanów

## Działanie algorytmu:



n – liczba określająca kolejność analizy wierzchołków dla algorytmu „w głąb”  
m – liczba określająca kolejność analizy wierzchołków dla algorytmu „wszerz”

### Szukanie z nawracaniem (BT)

Strategia z nawracaniem jest modyfikacją algorytmu przeszukiwania „w głąb”. Lista OPEN jest również zorganizowana w postaci stosu, jednak dla badanego wierzchołka generowany jest tylko jeden jego potomek.

Jeżeli nowy wierzchołek nie spełnia kryterium celu lub końcowego, to jest dalej rozszerzany (ale określamy tylko jednego potomka).

Gdy po kolejnych rozszerzeniach otrzymany wierzchołek jest końcowy lub nie można dla niego określić wierzchołka potomnego następuje powrót do najbliższego przodka, dla którego możliwe jest wygenerowanie potomków.

Główną zaletą strategii z powracaniem jest oszczędność pamięci – dla danego wężła wystarczy zapamiętać tylko jednego potomka.

Oszczędność pamięci okupiona jest jednak komplikacją algorytmu – przy powrocie do wcześniej analizowanego wierzchołka należy sprawdzić, które krawędzie z niego wychodzące zostały już rozpatrzone.

Jednym z problemów algorytmu szukania z nawracaniem jest możliwość powstawania pętli (powrót do stanów już raz zbadanych). Rozwiązaniem tego problemu może być utworzenie dodatkowej listy CLOSED (zawiera wykaz stanów już rozpatrzonych).