



Politechnika Wroclawska

# Heurystyczne metody przeszukiwania

Dariusz Banasiak

Katedra Informatyki Technicznej W4/K9

Politechnika Wroclawska

Metody heurystyczne są jednym z ważniejszych narzędzi sztucznej inteligencji.

### Pojęcie heurystyki

**Heurystyka** (z j. grec. *heuresis* – znalezienie, odkrycie) – sztuka znajdowania rozwiązań problemów lub odkrywania nowych metod ich rozwiązywania.

**Heurystyka** (wg Słownika języka polskiego) – umiejętność wykrywania nowych faktów i związków pomiędzy faktami (a zwłaszcza stawiania hipotez), prowadzącego do poznania nowych prawd.

### Cele stosowania metod heurystycznych:

- poprawa efektywności dla złożonych problemów
- poszukiwanie rozwiązania wzdłuż najkrótszej, najbardziej prawdopodobnej drogi (ominięcie mniej obiecujących ścieżek)
- formułowanie prostych kryteriów wyboru kierunków postępowania
- unikanie badania tak zwanych „ślepych uliczek” np. wykorzystanie zdobytych wcześniej informacji

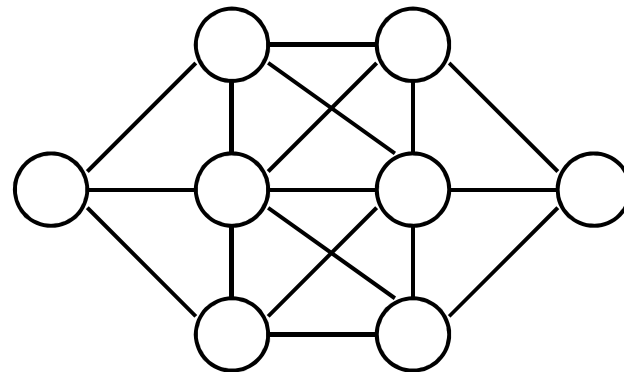
Metody heurystyczne stanowią najczęściej modyfikację algorytmu szukania „w głąb” polegającą na wprowadzeniu funkcji oceny przydatności poszczególnych „ruchów”, co umożliwia uporządkowanie wierzchołków na danym poziomie (i ich odpowiedni wybór).

Dla określenia heurystyki niezbędna jest dodatkowa wiedza dotycząca badanego problemu. Informacje dotyczące danego problemu nie zawsze są sformułowane ściśle i precyzyjnie (np. doświadczenie osoby opisującej dany problem).

**Określenie heurystyki dla konkretnego problemu jest często zadaniem dosyć trudnym!**

## Zadanie o 8 kółkach

W diagramie należy umieścić litery A, B, C, D, E, F, G i H w taki sposób, aby żadna litera nie sąsiadowała z literą występującą bezpośrednio przed lub po niej w alfabecie.



**Istnieje  $8! = 40320$  sposobów rozmieszczenia liter!**

## Przykład podejścia heurystycznego

Aby rozwiązać powyższy problem można zauważyć:

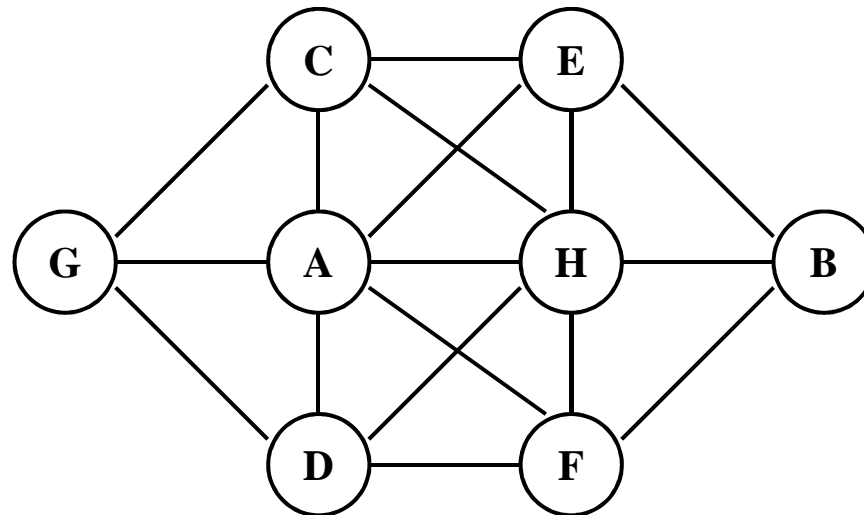
- najłatwiej jest umieścić litery A i H (posiadają tylko jedną literę, z którą nie mogą sąsiadować)
- najtrudniej wypełnić kółka w środku, gdyż każde sąsiaduje z sześcioma innymi

Rozwiązanie problemu:

- litery A i H należy umieścić w środkowych kółkach
- litery B i G mogą wystąpić tylko w tej samej linii poziomej co A i H (oczywiście B obok H, a G obok A)
- litery C i D muszą wystąpić w tej samej kolumnie co A, natomiast E i F w tej samej kolumnie co H

## Przykład podejścia heurystycznego

Ostateczne rozwiązanie:



### Metody przeszukiwania heurystycznego:

- szukanie typu „wspinanie do góry” (**HC** ang. hill climbing),
- szukanie „najpierw najlepszy” (**BFS** ang. best-first search),
- algorytm A\*,
- algorytm A\* z iteracyjnym poprawianiem,
- szukanie z ograniczeniem kosztu (**UCS** ang. uniform cost search),
- algorytm stopniowego studzenia, zwany wyżarzaniem (ang. simulated annealing)

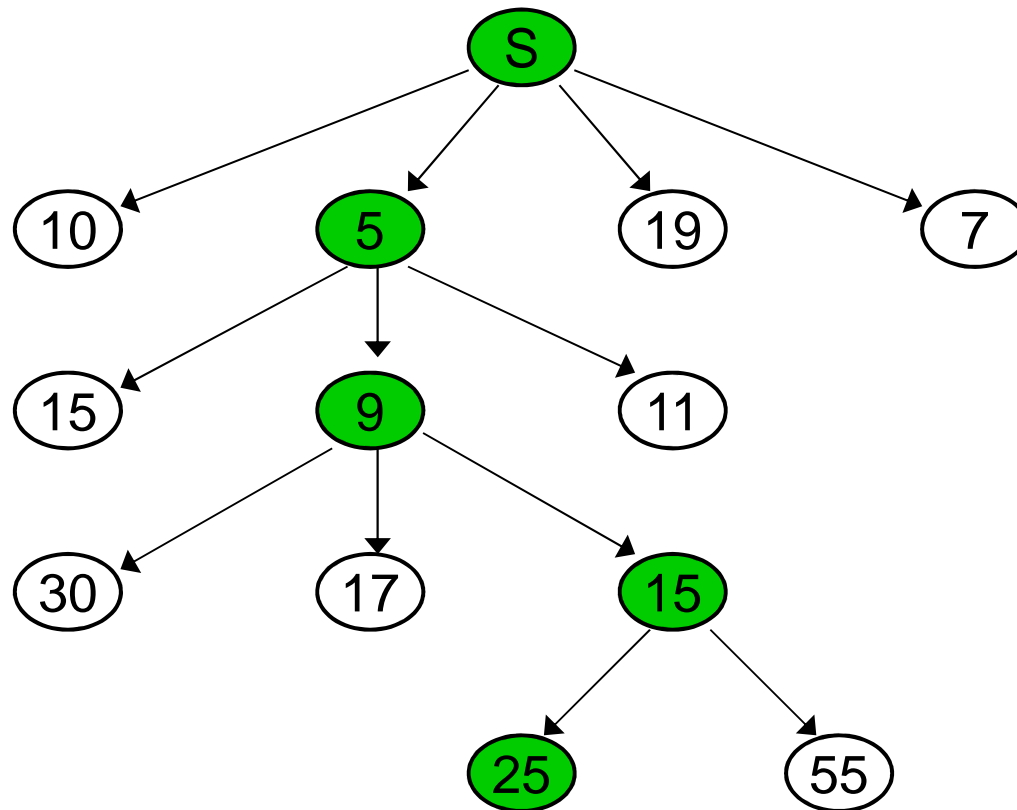


### Metoda „wspinanie do góry”

Metoda przeszukiwania typu „wspinanie do góry” jest najprostszym sposobem przeszukiwania heurystycznego. Jest to metoda gradientowa, która wybiera na każdym kroku maksymalnie korzystne rozwiązania.

Ponieważ nie przewiduje ona powrotów, nie gwarantuje ona uzyskania optymalnego rozwiązania (rozwiązanie najlepsze na pewnym etapie, nie musi być najlepsze jako ostateczne).

Przeszukiwanie grafu metodą „wspinanie do góry”:



### Metoda „najpierw najlepszy”

Sposób wyboru następnego wierzchołka do badania w metodzie „najpierw najlepszy” wydaje się być najbardziej naturalny. Do dalszego rozszerzania grafu poszukiwań wybierany jest najlepszy węzeł spośród wszystkich węzłów rozpatrywanych do tej pory, niezależnie od ich położenia w grafie.

Strategię tą można uznać za kombinację dwóch metod: poszukiwania „wszerz” i poszukiwania „w głąb”.

### Algorytm dla metody „najpierw najlepszy”:

1. wstaw wierzchołek początkowy na listę OPEN i oblicz  $g(S)$
2. IF (lista OPEN jest pusta) THEN STOP (brak rozwiązania)
3.  $St := \text{FIRST}(\text{OPEN})$
4. IF (St jest wierzchołkiem docelowym) THEN STOP (sukces)
5. usuń St z listy OPEN
6. utwórz wszystkie wierzchołki potomne od St, policz dla każdego z tych wierzchołków funkcję  $g(S_i)$ , wstaw te wierzchołki na listę OPEN i uporządkuj je rosnąco,
7. GOTO 2

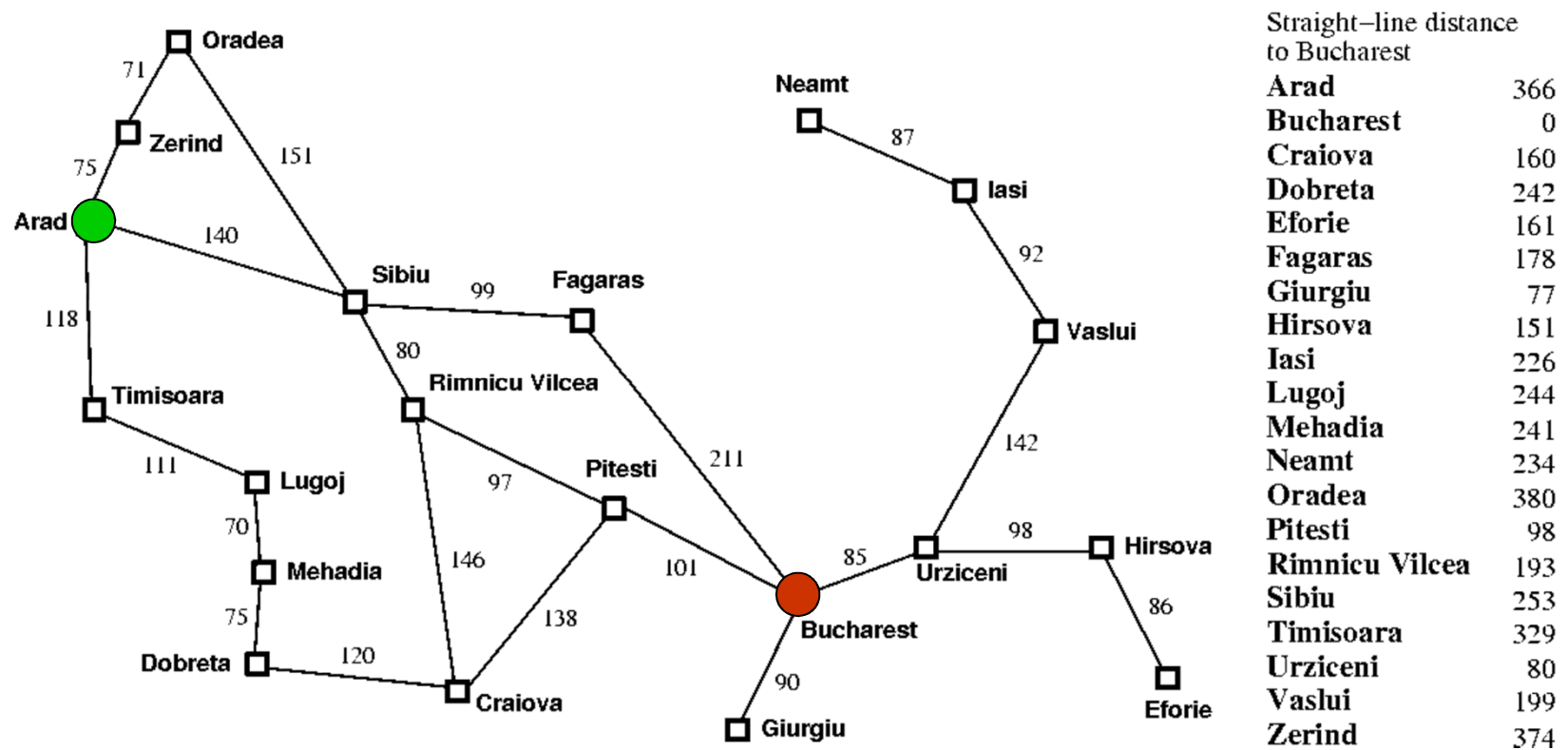
S – wierzchołek początkowy

$g(S_i)$  – heurystyczna funkcja oceny wierzchołka  $S_i$

# Heurystyczne metody przeszukiwania

## Przykład 1

Należy znaleźć najkrótszą drogę z miejscowości **Arad** do **Bukaresztu**



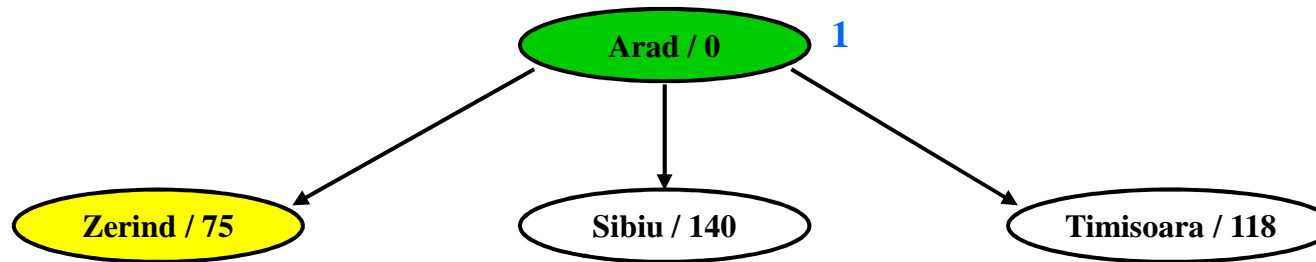
Budowę drzewa poszukiwań rozpoczynamy od rozwinięcia węzła początkowego. W wyniku otrzymujemy trzy nowe wierzchołki: Zerind, Sibiu i Timisoara.

Jako funkcję oceny danego wierzchołka przyjęto jego odległość od wierzchołka początkowego. Za każdym razem rozwijany jest ten wierzchołek, któremu przypisano najmniejszą wartość funkcji oceny (bez względu na jego położenie w grafie).

W związku z tym kolejnym rozwijanym wierzchołkiem będzie Zerind (funkcja oceny równa 75).

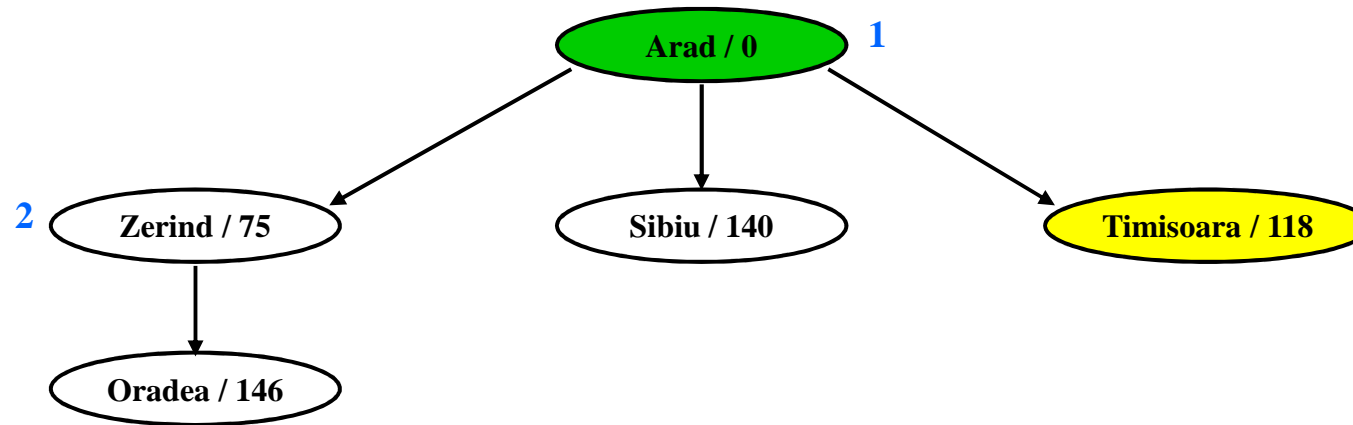
## Heurystyczne metody przeszukiwania

Kolejne etapy budowy drzewa poszukiwań (krok 1):



## Heurystyczne metody przeszukiwania

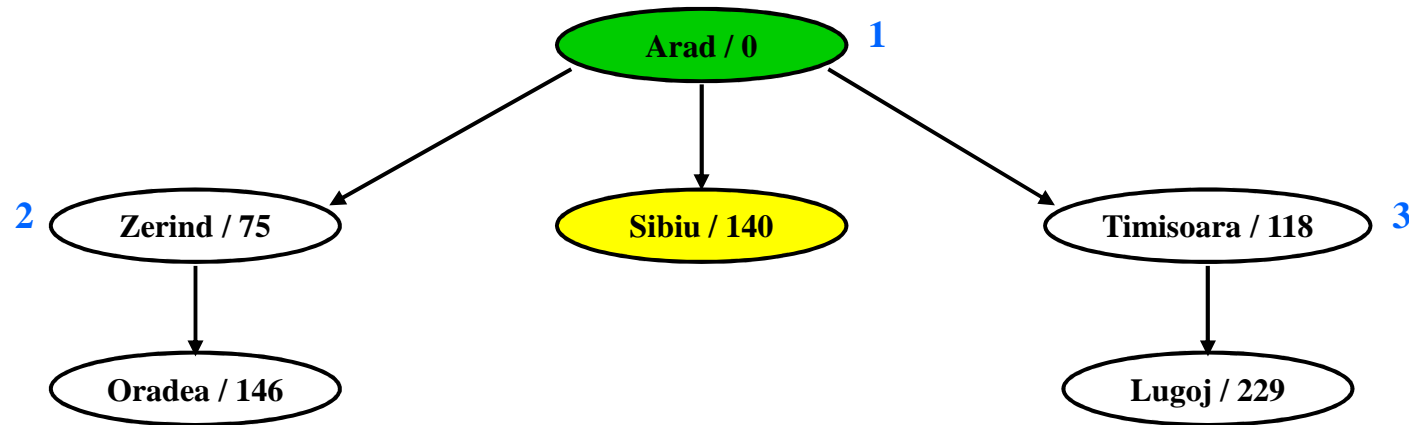
Kolejne etapy budowy drzewa poszukiwań (krok 2):





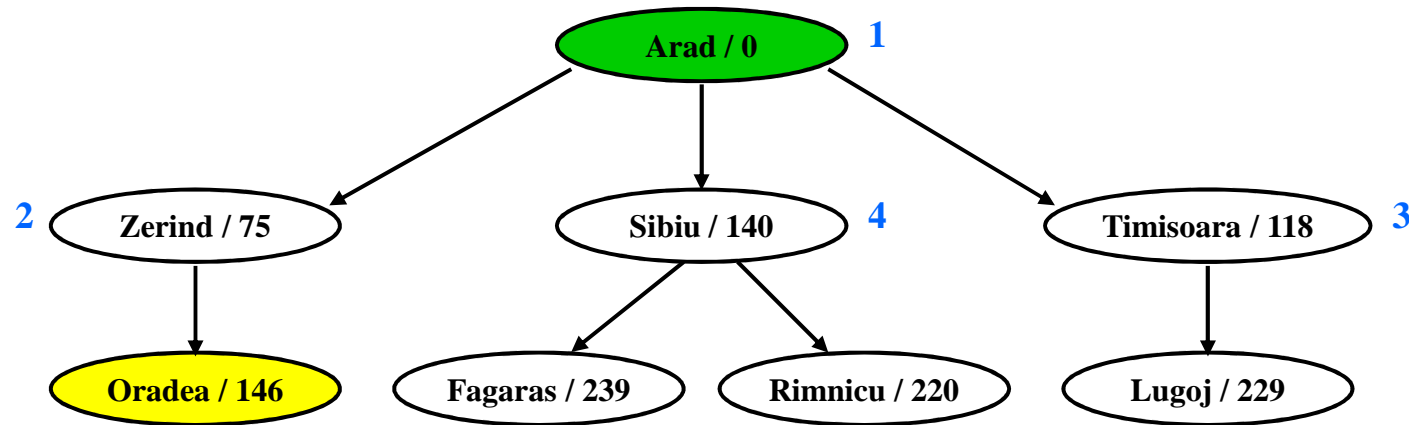
# Heurystyczne metody przeszukiwania

Kolejne etapy budowy drzewa poszukiwań (krok 3):



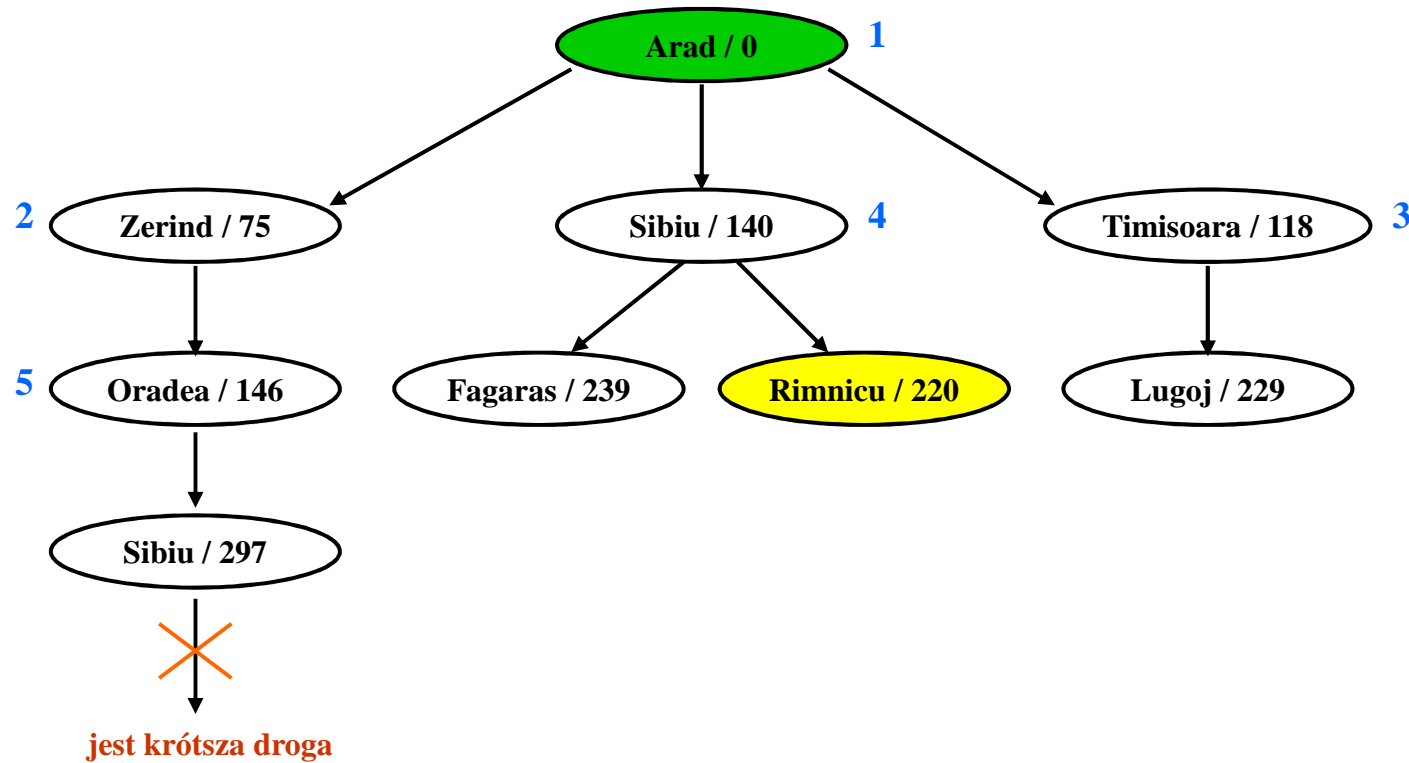
# Heurystyczne metody przeszukiwania

Kolejne etapy budowy drzewa poszukiwań (krok 4):



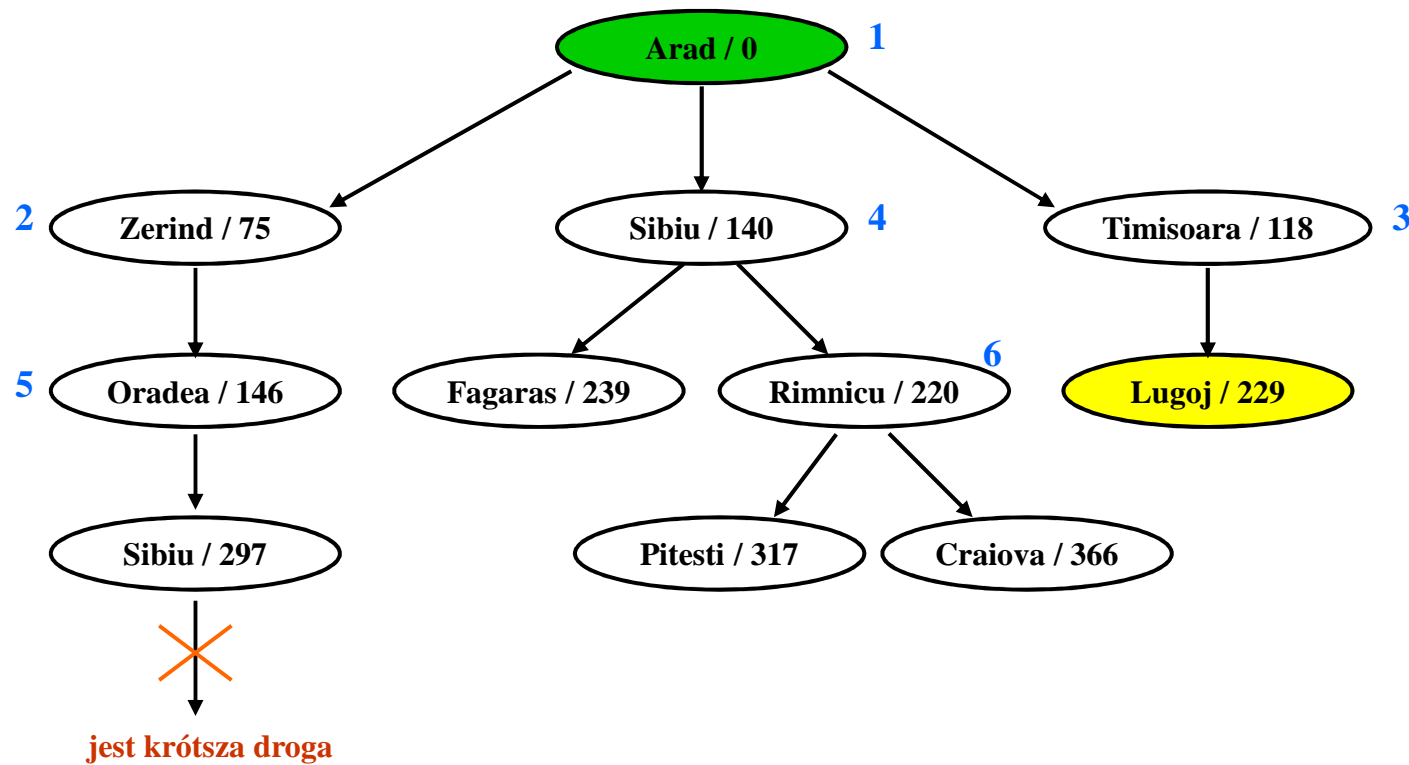
# Heurystyczne metody przeszukiwania

Kolejne etapy budowy drzewa poszukiwań (krok 5):



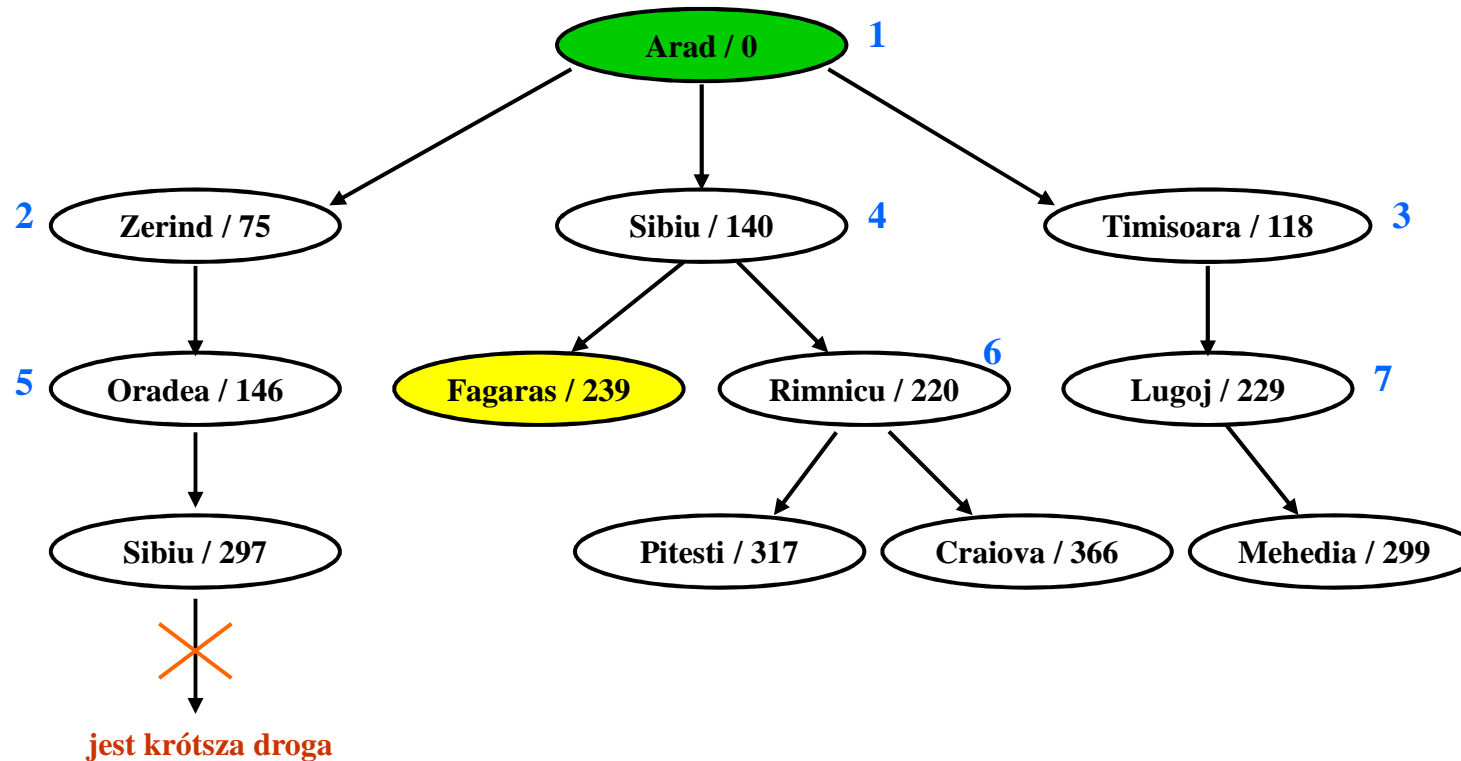
# Heurystyczne metody przeszukiwania

Kolejne etapy budowy drzewa poszukiwań (krok 6):



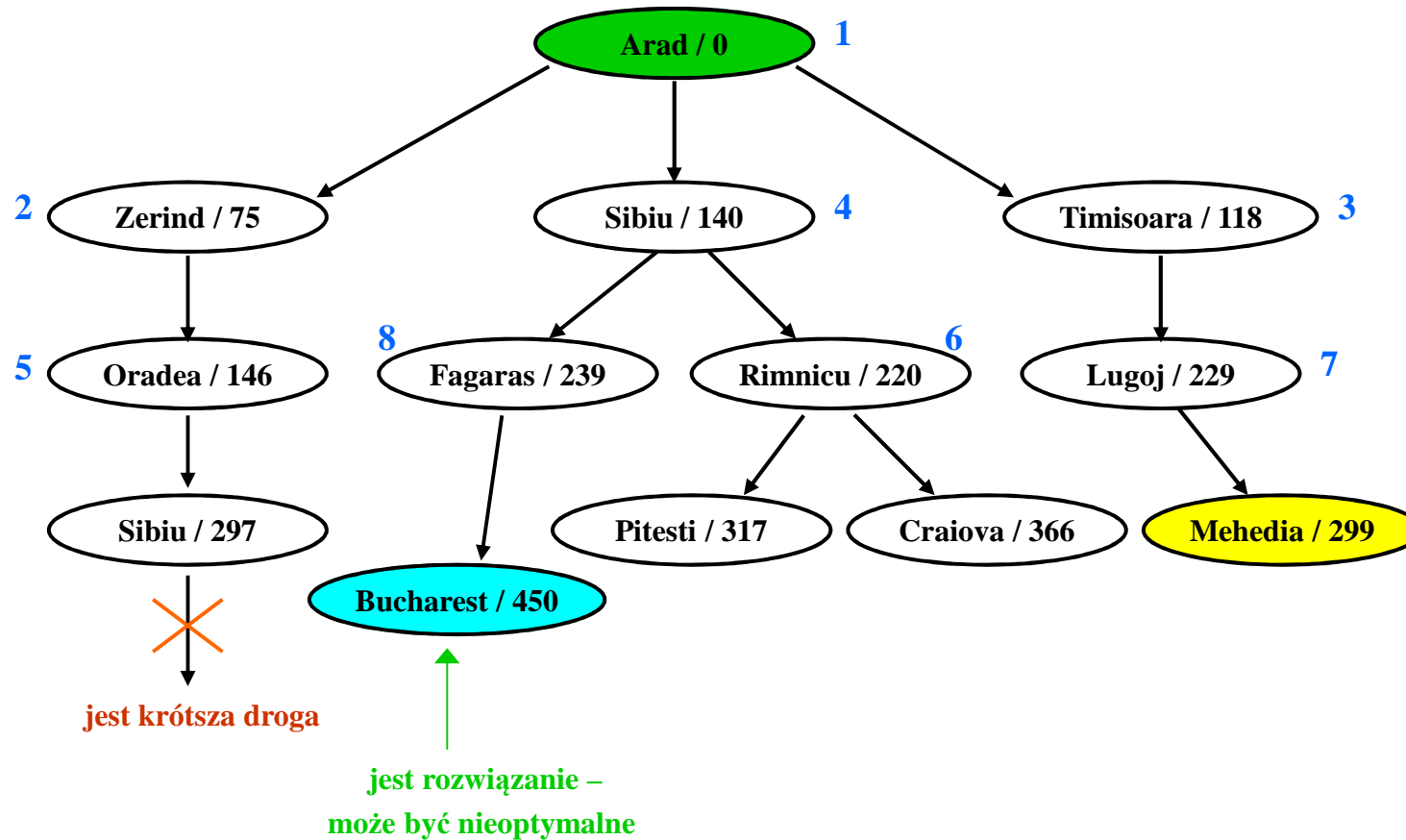
# Heurystyczne metody przeszukiwania

Kolejne etapy budowy drzewa poszukiwań (krok 7):



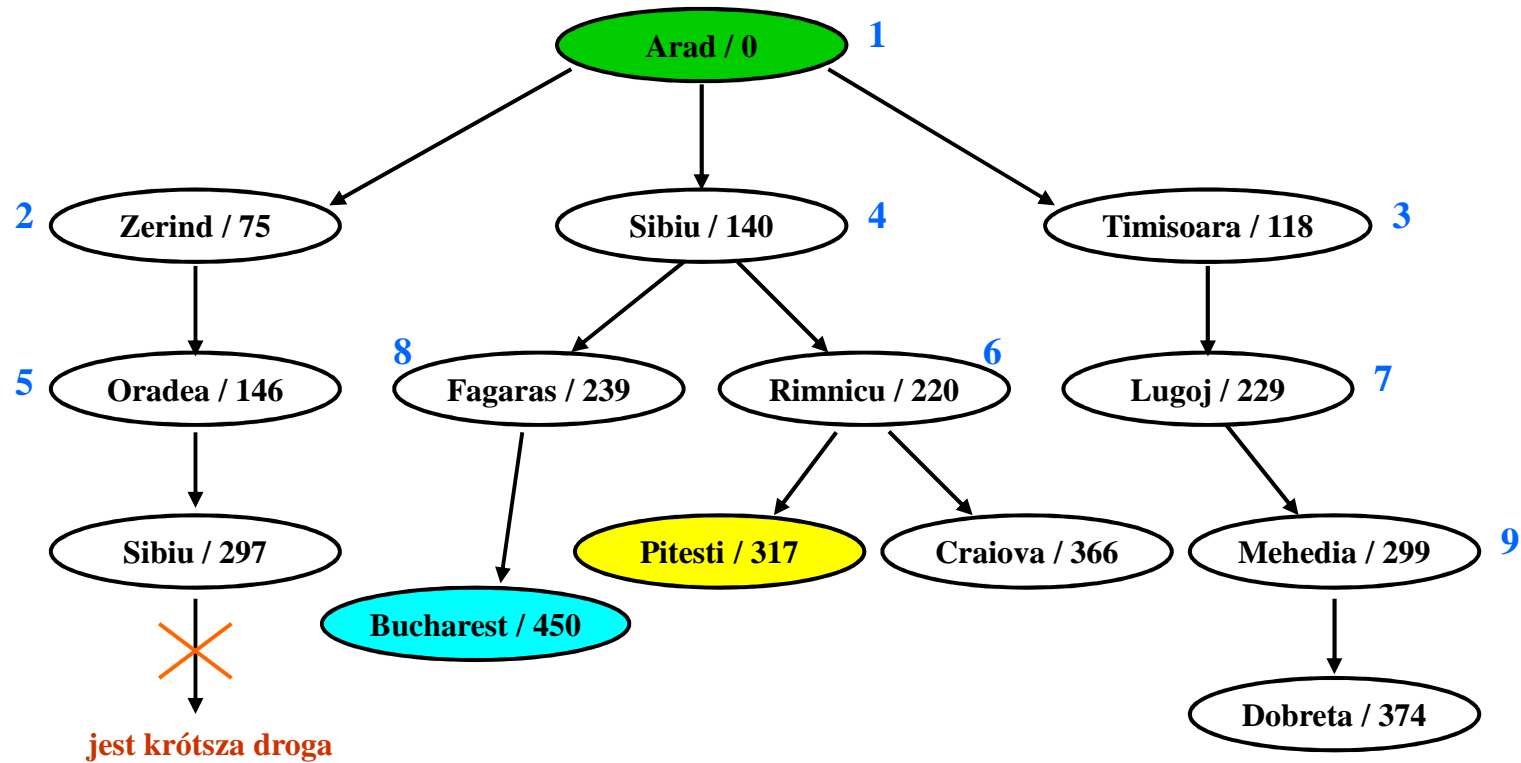
# Heurystyczne metody przeszukiwania

Kolejne etapy budowy drzewa poszukiwań (krok 8):



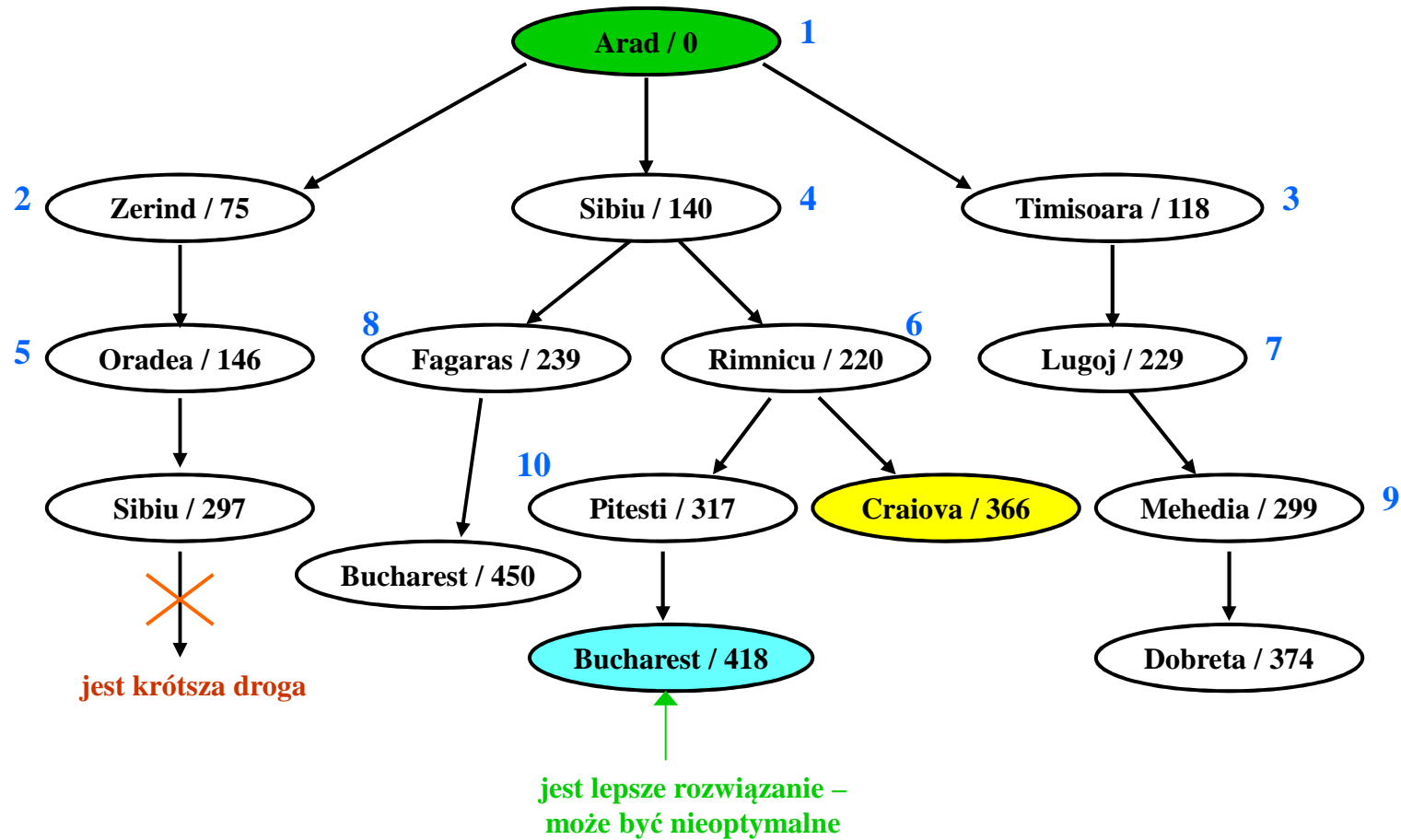
# Heurystyczne metody przeszukiwania

Kolejne etapy budowy drzewa poszukiwań (krok 9):



# Heurystyczne metody przeszukiwania

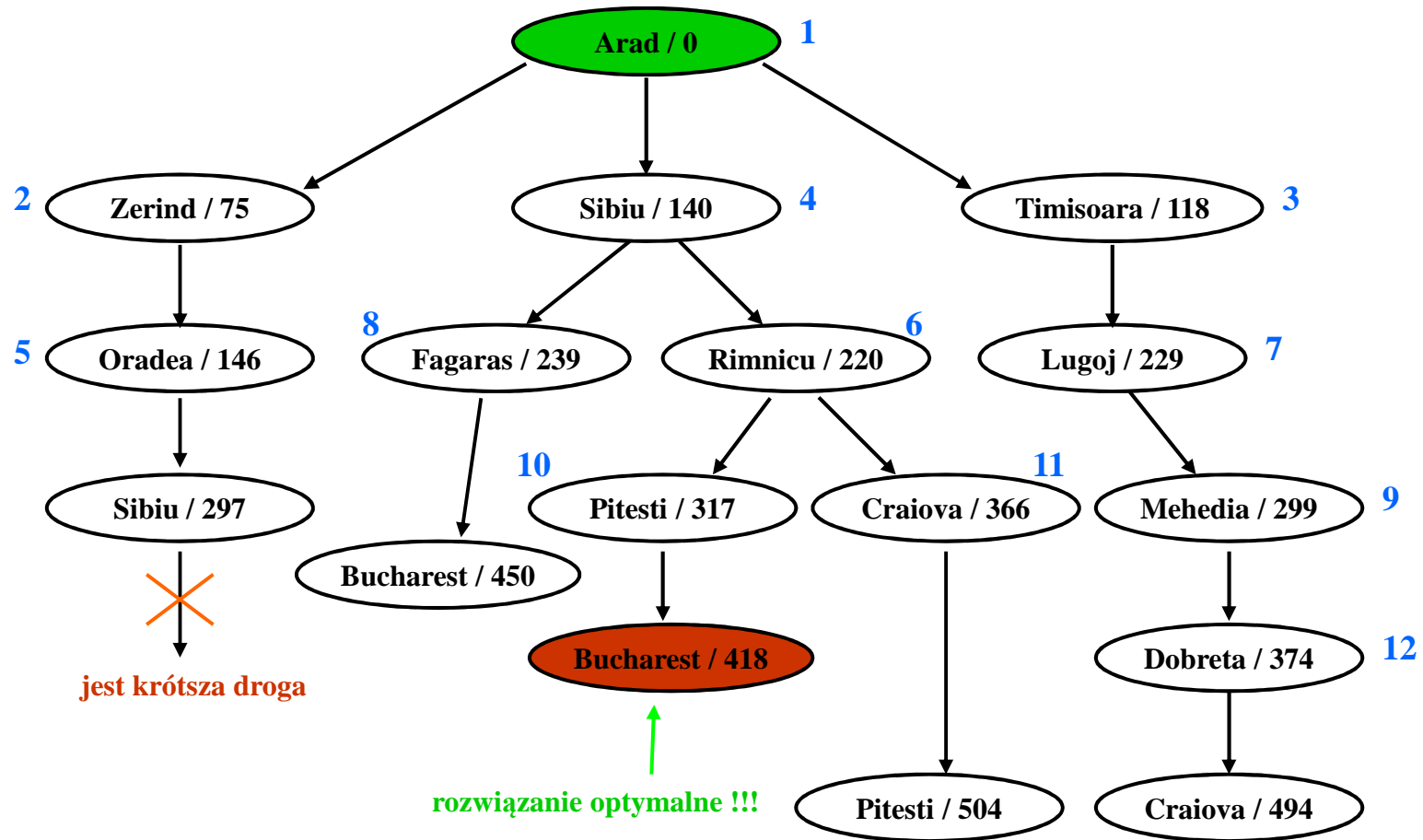
Kolejne etapy budowy drzewa poszukiwań (krok 10):





# Heurystyczne metody przeszukiwania

Kolejne etapy budowy drzewa poszukiwań (kroki 11 i 12):



Jedną z wersji metody „najpierw najlepszy” jest przeszukiwanie zachłanne (GS – ang. Greedy Search). W metodzie tej ocenie podlega odległość danego wierzchołka od wierzchołka końcowego.

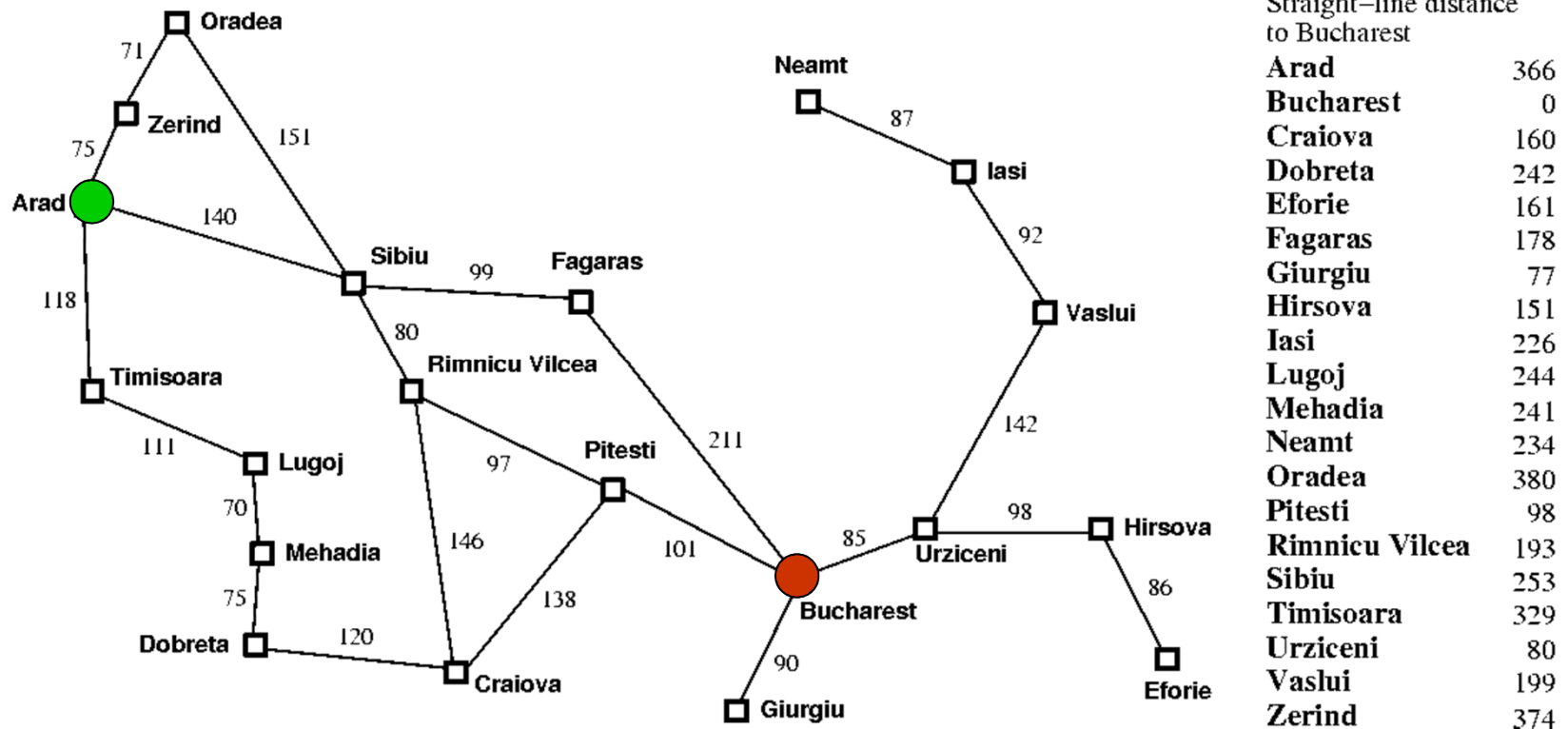
**W ogólnym przypadku skonstruowanie odpowiedniej funkcji heurystycznej nie jest zadaniem prostym!!!**

Dla naszego przypadku jako funkcję oceny wierzchołka (miasta) przyjmuje się jego odległość w linii prostej do wierzchołka docelowego (Bukareszt).

Jedną z metod tworzenia funkcji heurystycznych jest rozważenie problemu uproszczonego, w którym rezygnuje się ze spełnienia trudnych wymagań.

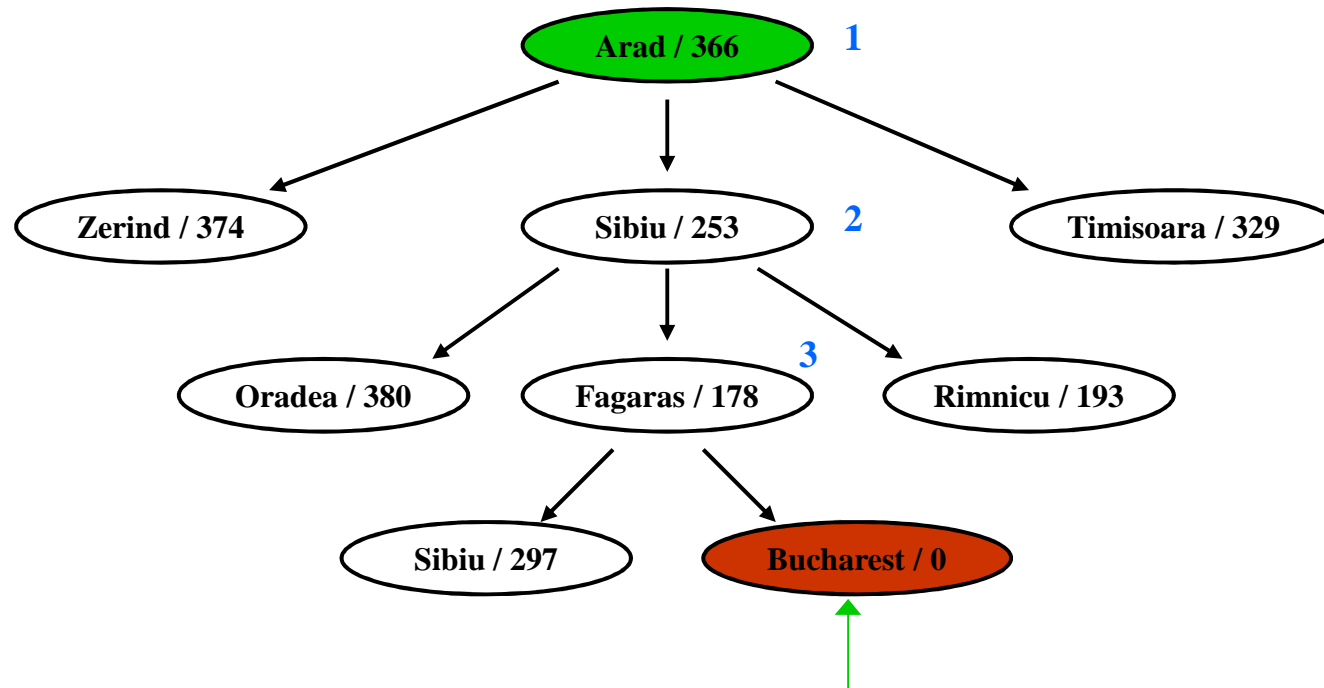
# Heurystyczne metody przeszukiwania

Należy znaleźć najkrótszą drogę z miejscowości **Arad** do **Bukaresztu**



# Heurystyczne metody przeszukiwania

Rozwiązanie problemu metodą przeszukiwania zachłannego:



znalezione rozwiązanie jest nieoptymalne (koszt wynosi 450),  
ale zostało ono znalezione w wyniku mniejszej liczby kroków !!!

### Przykład 2 – Problem 5 hetmanów

Na 25 polowej szachownicy należy rozmieścić pięciu hetmanów w taki sposób, aby żaden z hetmanów nie atakował innego.

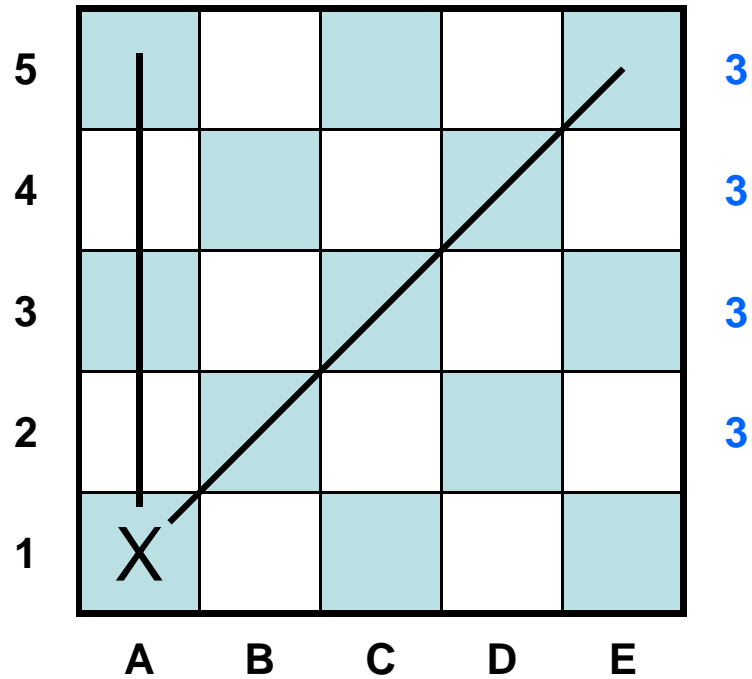
Poczynając od pustej szachownicy figury hetmana umieszczane są w kolejnych wierszach o numerach od 1 do 5.

Do oceny aktualnej pozycji wykorzystywana jest funkcja heurystyczna, której wartość określa minimalną liczbę nie atakowanych pól w wolnych wierszach.

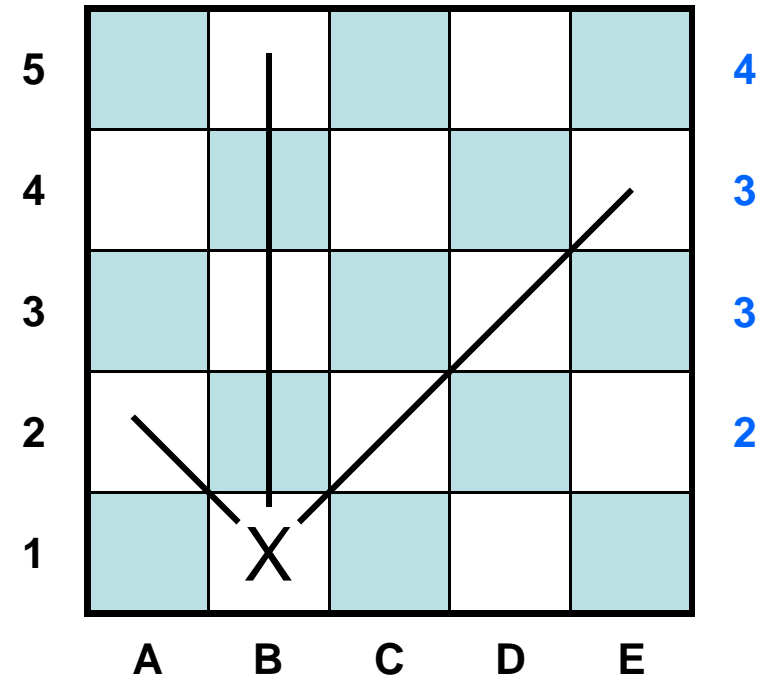
Ponieważ przyjmuje się, że kolejność węzłów ustalana jest według minimalnych wartości funkcji heurystycznej, wartości oceny poszczególnych wierzchołków są liczbami ujemnymi.

# Heurystyczne metody przeszukiwania

Obliczenie wartości funkcji heurystycznej:



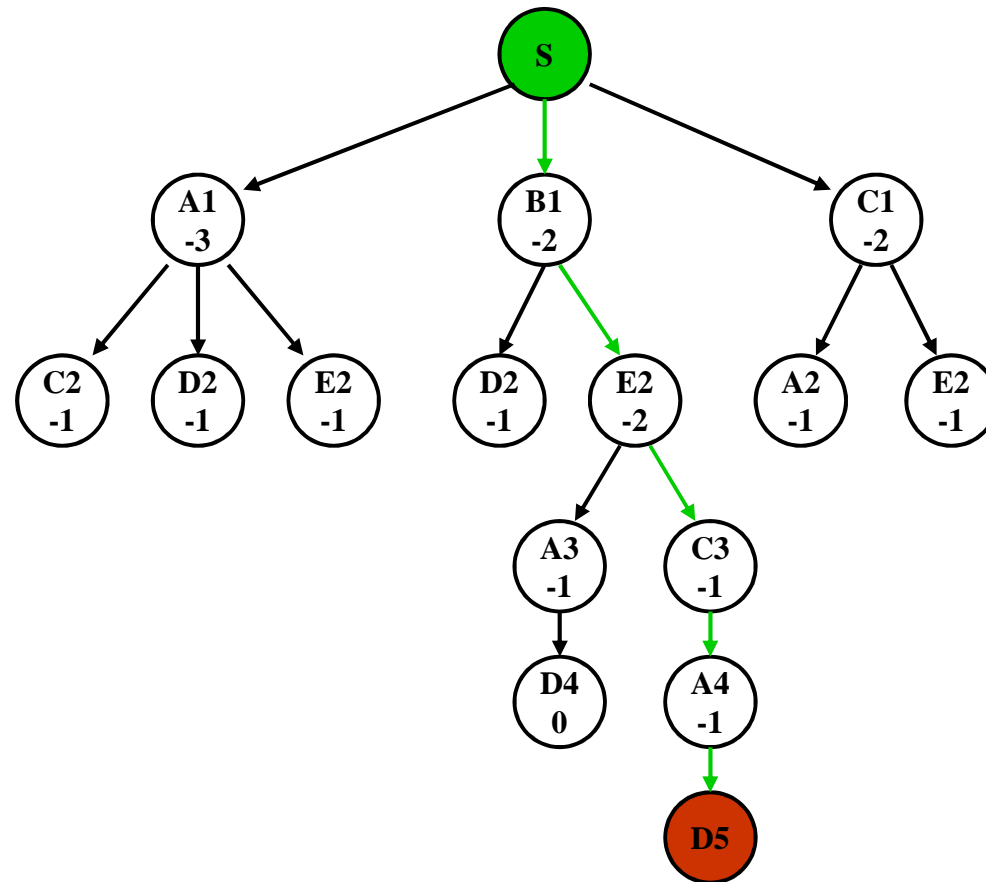
$$g(A1) = - \min(3, 3, 3, 3) = -3$$



$$g(B1) = - \min(2, 3, 3, 4) = -2$$

# Heurystyczne metody przeszukiwania

Drzewo poszukiwań dla problemu 5 hetmanów:



Ostateczne rozwiązanie problemu:

5			X		
4	X				
3		X			
2				X	
1	X				
	A	B	C	D	E



### Algorytm A\*

Bardzo ważną metodą heurystycznego poszukiwania jest strategia A\*. Stanowi ona połączenie przedstawionych wcześniej metod konstrukcji funkcji oceny wierzchołków. Funkcja heurystyczna w strategii A\* jest sumą dwóch składników:

$$f^*(n) = g^*(n) + h^*(n)$$

$g^*(n)$  – koszt drogi z wierzchołka początkowego S do wierzchołka n,

$h^*(n)$  – koszt drogi z wierzchołka n do wierzchołka docelowego T.

Wybór kolejnego wierzchołka odbywa się zgodnie ze strategią „najpierw najlepszy”. Główny problem stanowi wybór funkcji heurystycznej  $h^*(n)$ , której zadaniem jest ocena odległości danego wierzchołka od stanu końcowego.

Przy konstrukcji tej funkcji należy więc wykorzystać wszelką dostępną wiedzę dotyczącą problemu (często stosuje się analogię do już rozwiązanych problemów).

Czasami wykorzystywane są pomysły, które nie posiadają formalnego uzasadnienia.

Przykładowe heurystyki dla problemu puzzli o 9 polach:

- $h_1(n) = W(n)$ , gdzie  $W(n)$  oznacza ilość elementów, które nie znajdują się na swoim miejscu
- $h_2(n) = P(n)$ , gdzie  $P(n)$  oznacza sumę odległości wszystkich elementów, które nie znajdują się na swoim miejscu od ich właściwego położenia,
- $h_3(n) = P(n) + 3 \cdot S(n)$ , gdzie  $P(n)$  – jak wyżej,  $S(n)$  natomiast jest obliczana dla elementów znajdujących się na obrzeżu (do funkcji tej dodajemy 2 dla każdego elementu, po którym następuje niewłaściwy element oraz dodatkowo 1, jeżeli występuje element na środku).

## Heurystyczne metody przeszukiwania

$d$	Search Cost			Effective Branching Factor		
	IDS	$A^*(h_1)$	$A^*(h_2)$	IDS	$A^*(h_1)$	$A^*(h_2)$
2	10	6	6	2.45	1.79	1.79
4	112	13	12	2.87	1.48	1.45
6	680	20	18	2.73	1.34	1.30
8	6384	39	25	2.80	1.33	1.24
10	47127	93	39	2.79	1.38	1.22
12	3644035	227	73	2.78	1.42	1.24
14	–	539	113	–	1.44	1.23
16	–	1301	211	–	1.45	1.25
18	–	3056	363	–	1.46	1.26
20	–	7276	676	–	1.47	1.27
22	–	18094	1219	–	1.48	1.28
24	–	39135	1641	–	1.48	1.26

**Figure 4.8** Comparison of the search costs and effective branching factors for the ITERATIVE-DEEPENING-SEARCH and  $A^*$  algorithms with  $h_1$ ,  $h_2$ . Data are averaged over 100 instances of the 8-puzzle, for various solution lengths.