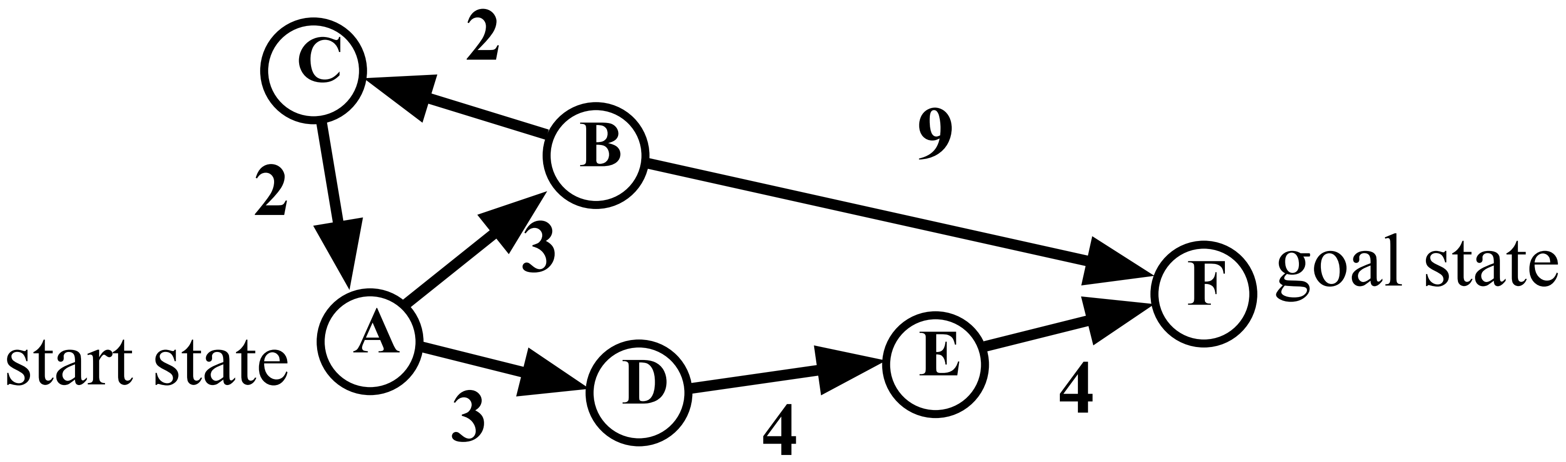
Informed
/Heuristic Search

Informed search

- Uses problem-specific knowledge beyond the definition of the problem itself
- Can find solutions more efficiently than an uninformed strategy.
- Expand closer-seeming nodes first

Heuristics

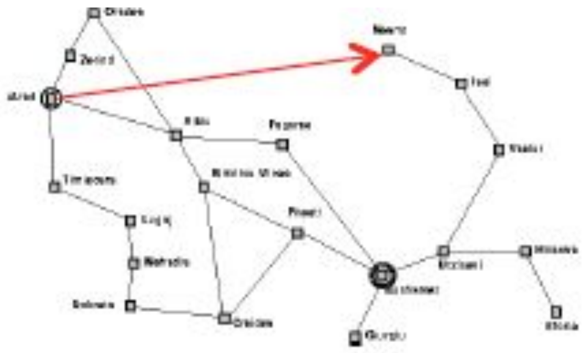
- Key notion: heuristic function h(n) gives an estimate of the distance from n to the goal
  - h(n)=0 for goal nodes
- E.g. straight-line distance for traveling problem

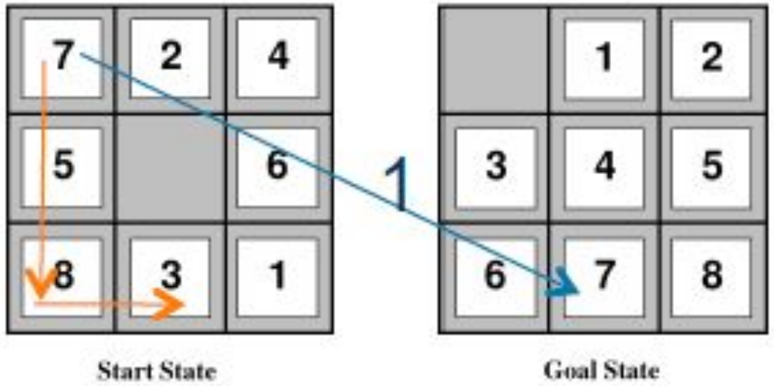


- Say: h(A) = 9, h(B) = 8, h(C) = 9, h(D) = 6, h(E) = 3, h(F) = 0
- We're adding something new to the problem!
- Can use heuristic to decide which nodes to expand first

Examples - Heuristics

- Travel planning
  - **Euclidean distance**
- 8-puzzle
  - **Manhattan distance**
  - **Number of misplaced tiles**
- Traveling salesman problem
  - **Minimum spanning tree**





Best First Search

- Best-first search is an instance of the general TREE-SEARCH or GRAPH-SEARCH algorithm in which a node is selected for expansion based on an **evaluation function,** f (n)
- Best-first search can be implemented within our general search framework via a priority queue, a data structure that will maintain the fringe in ascending order of f -values.
- If the evaluation function is exactly accurate, then this will indeed be the best node

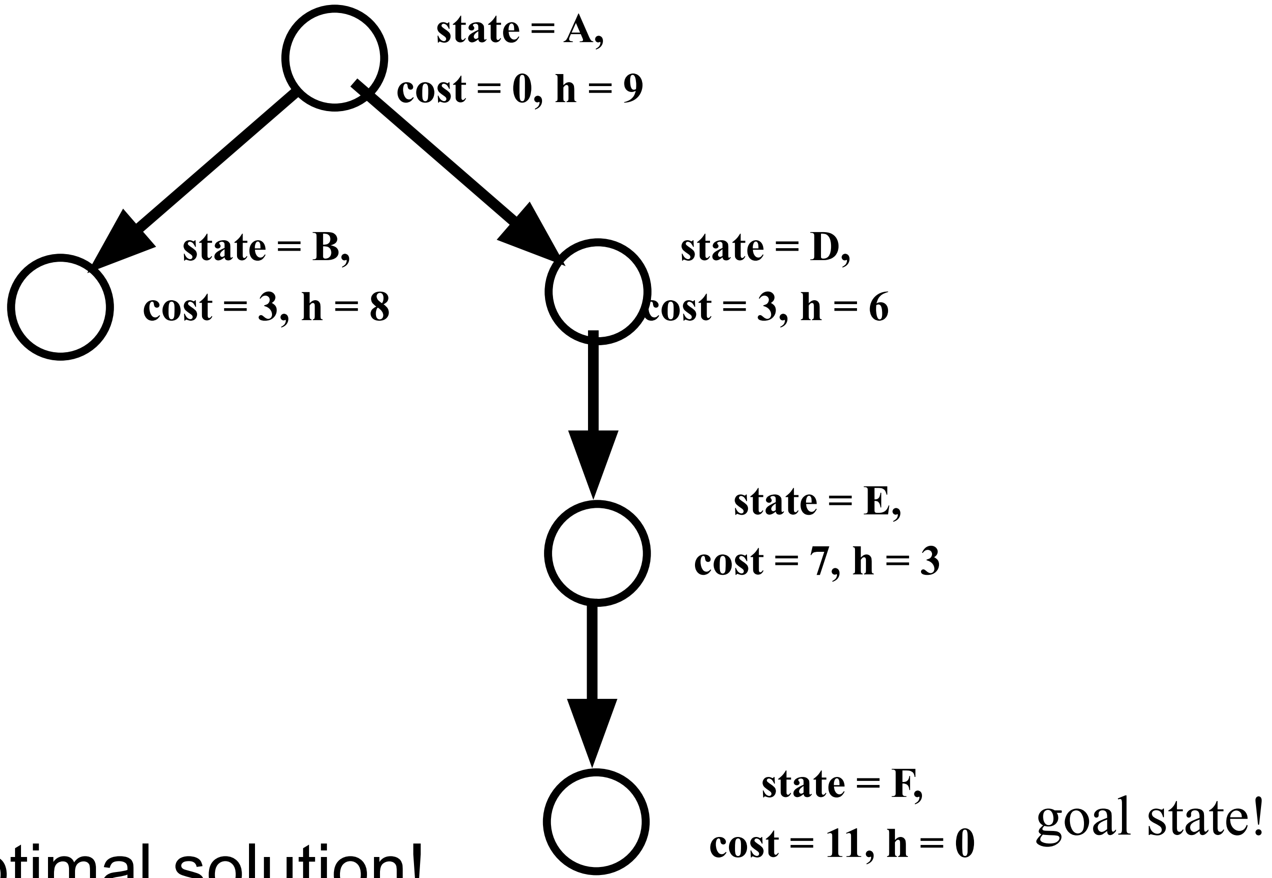
Best First Search

- There is a whole family of BEST-FIRST-SEARCH algorithms with different evaluation functions.
- A key component of these algorithms is a heuristic function *h(n)*

  h(n) = estimated cost of the cheapest path from node *n* to a goal node
- Heuristic functions are the most common form in which additional knowledge of the problem is imparted to the search algorithm
- if *n* is a goal node, then *h(n)* = 0.

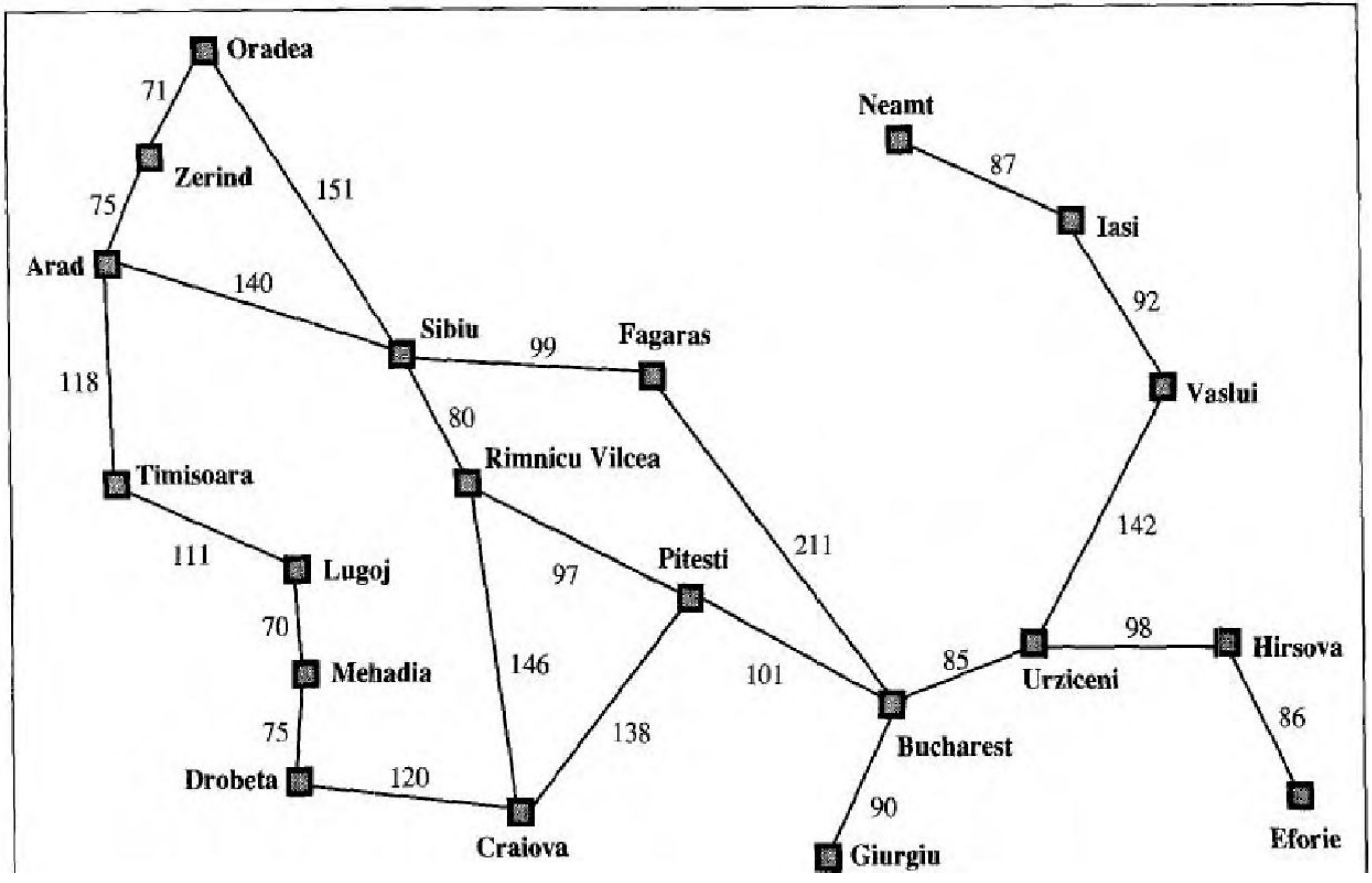
Greedy best-first search

- Thus, it evaluates nodes by using just the heuristic function: f (n) = h(n).
- Greedy best-first search: expand nodes with lowest h values first

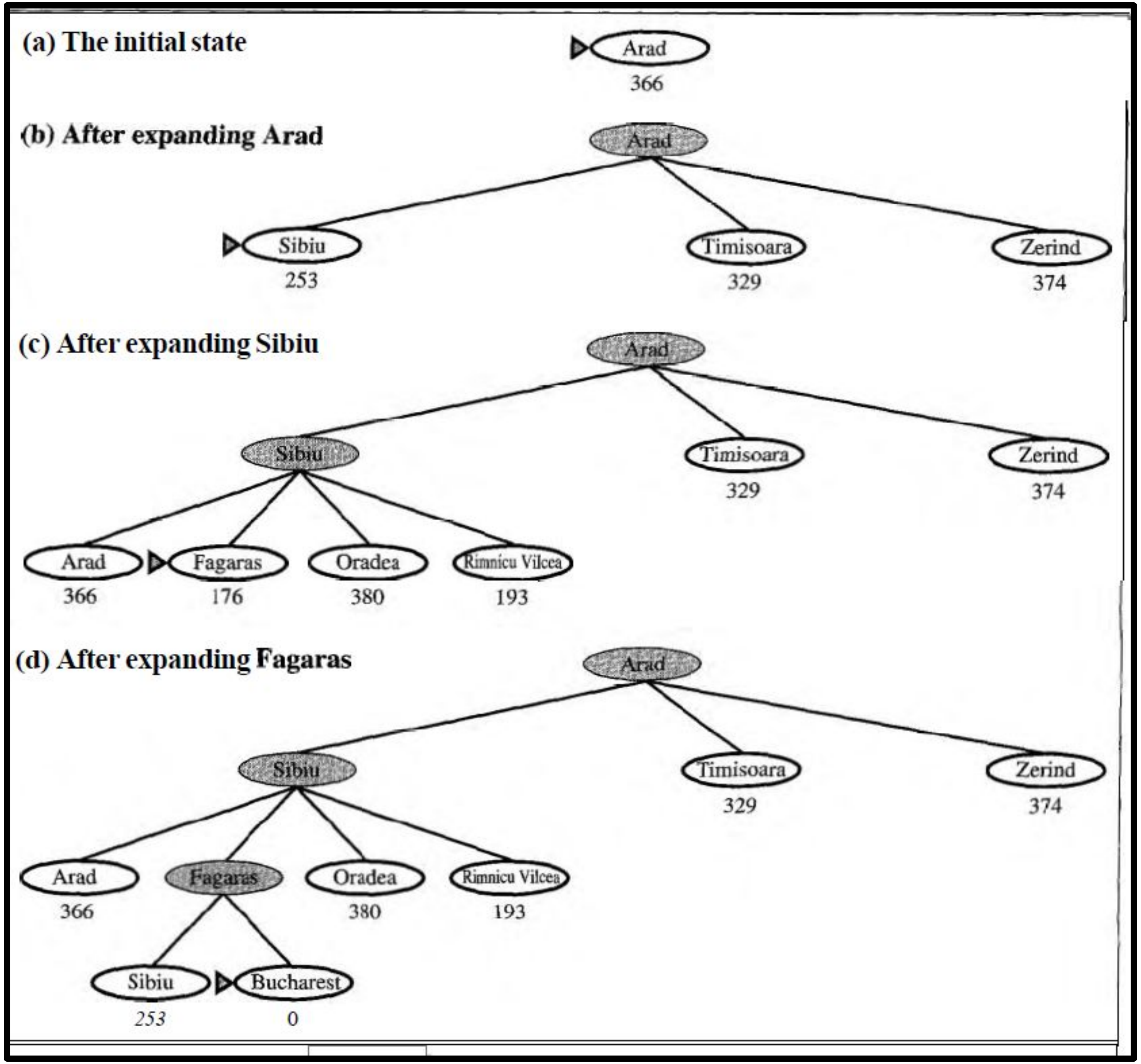


- Rapidly finds the optimal solution!
- Does it always?

A simplified road map of part of Romania.



Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Drobeta	242	**Pitesti**	100
Eforie	161	**Rimnicu Vilcea**	193
Fagaras	176	**Sibiu**	253
Giurgiu	77	**Timisoara**	329
Hirsova	151	**Urziceni**	80
Iasi	226	**Vaslui**	199
Lugoj	244	**Zerind**	374



(a) The initial state
Arad
366
(b) After expanding Arad
Arad
Sibiu
253
Timisoara
329
Zerind
374
(c) After expanding Sibiu
Arad
Sibiu
Timisoara
329
Zerind
374
Arad
366
Fagaras
176
Oradea
380
Rimnicu Vilcea
193
(d) After expanding Fagaras
Arad
Sibiu
Timisoara
329
Zerind
374
Arad
366
Fagaras
Oradea
380
Rimnicu Vilcea
193
Sibiu
253
Bucharest
0

A* search:

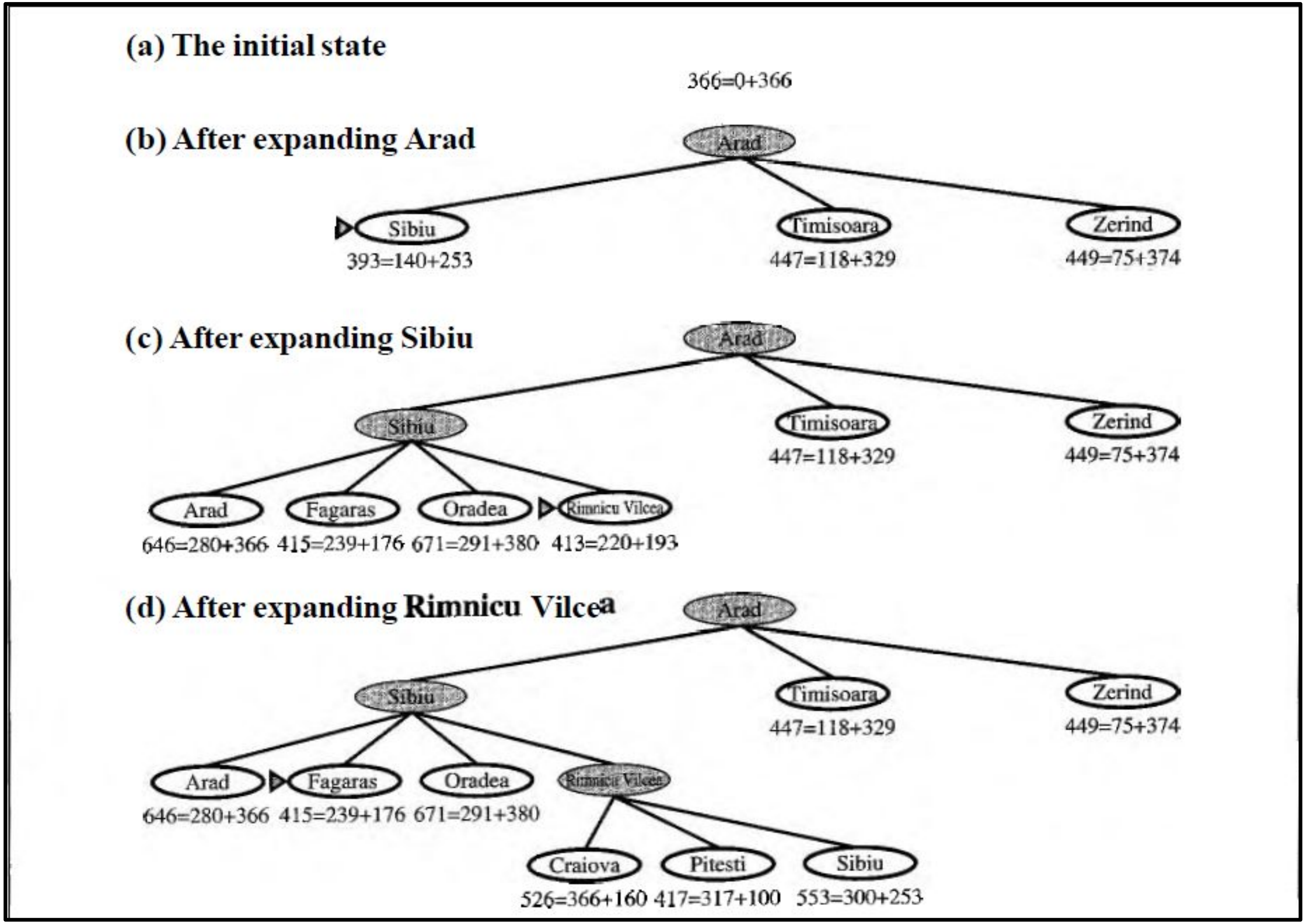
Minimizing the total estimated solution cost

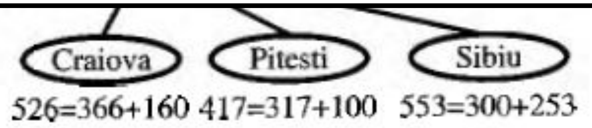
- The most widely-known form of best-first search is called **A*** search
- It evaluates nodes by combining *g(n),* the cost to reach the node, and *h(n.),* the cost to get from the node to the goal

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

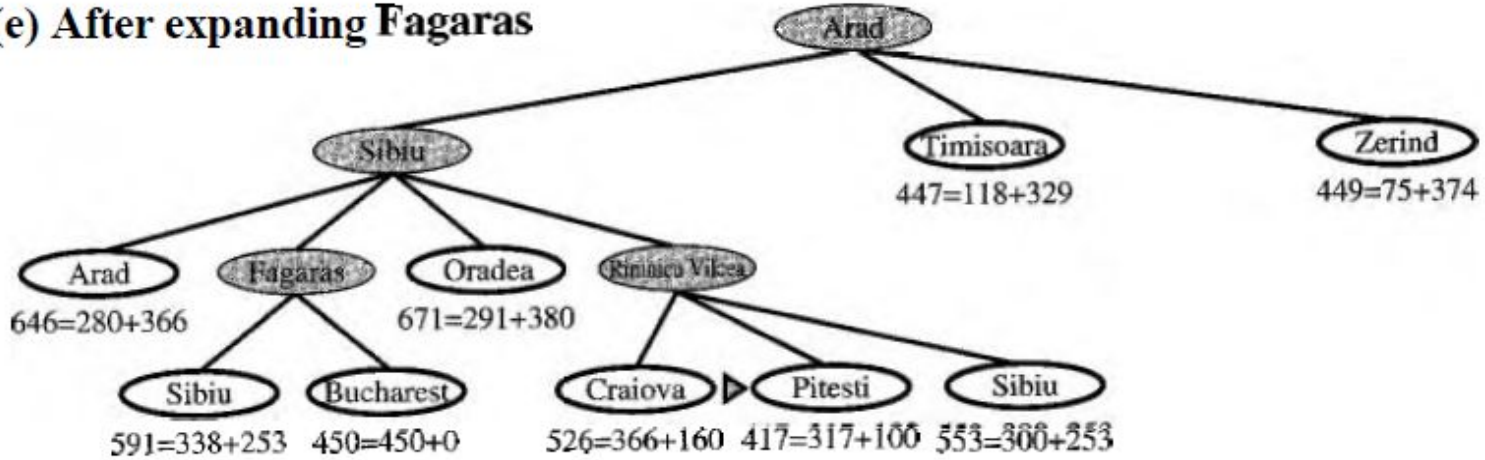
g(n) □ gives the path cost from the start node to node n, and

h(n) □ the estimated cost of the cheapest path from n to the goal,

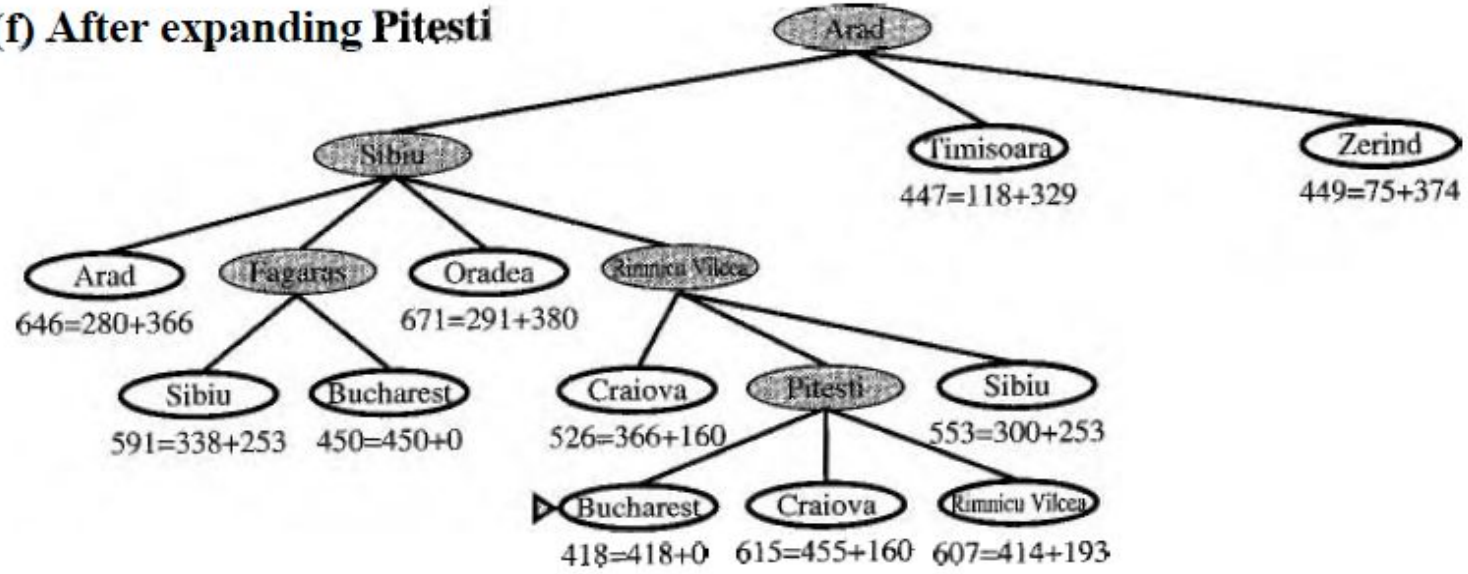

(a) The initial state
366=0+366
(b) After expanding Arad
Arad
Sibiu
393=140+253
Timisoara
447=118+329
Zerind
449=75+374
(c) After expanding Sibiu
Arad
Sibiu
Timisoara
447=118+329
Zerind
449=75+374
Arad
646=280+366
Fagaras
415=239+176
Oradea
671=291+380
Rimnicu Vilcea
413=220+193
(d) After expanding Rimnicu Vilcea
Arad
Sibiu
Timisoara
447=118+329
Zerind
449=75+374
Arad
646=280+366
Fagaras
415=239+176
Oradea
671=291+380
Rimnicu Vilcea
Craiova
526=366+160
Pitesti
417=317+100
Sibiu
553=300+253



(e) After expanding Fagaras

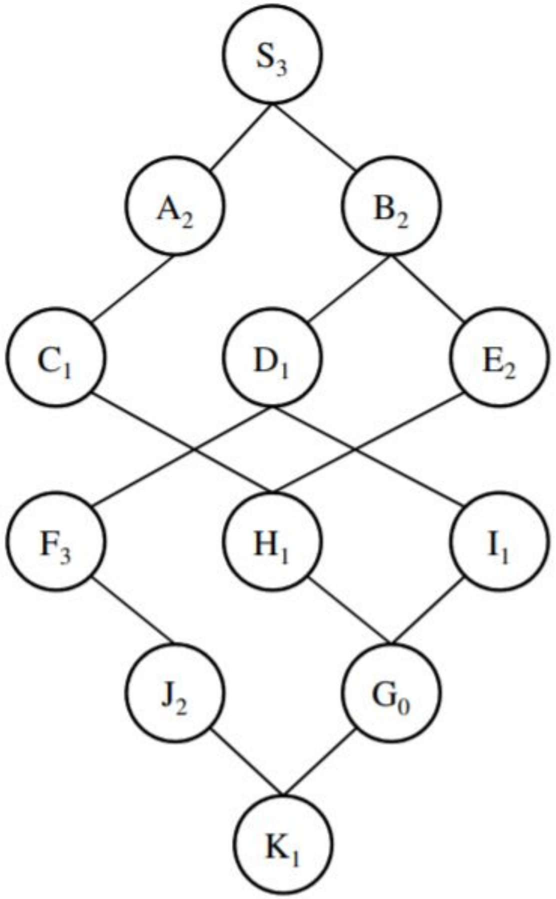


(f) After expanding Pitesti



Example

S_3	H_1	D_1	K_1
I_1	J_2	A_2	E_2
C_1	B_2	G_0	F_3



Depth-first search.

Breadth-first search.

S (0+3=3), A (1+2=3), B (1+2=3), C (2+1=3), D (2+1=3), E (2+2=4), H (3+1=4), G (4+0=4)

A*

The letter in each node is its name and the subscript digit is the heuristic value. All transitions have cost 1.

Admissible heuristics

A heuristic $h(n)$ is admissible if

for every node n,

$$h(n) \le h^*(n)$$

where $h^*(n)$ is the true cost to reach the goal state from n.

An admissible heuristic never overestimates the cost to reach the goal, i.e., it is optimistic

Is the Straight Line Distance heuristic $h_{SLD}(n)$ *admissible?*

Yes, it never overestimates the actual road distance

Admissibility

- A heuristic is admissible if it never overestimates the distance to the goal. i.e, If n is the optimal solution reachable from n’,

  then $g(n) \geq g(n') + h(n')$

- A* is admissible if it uses an admissible heuristic, and h(goal) = 0.

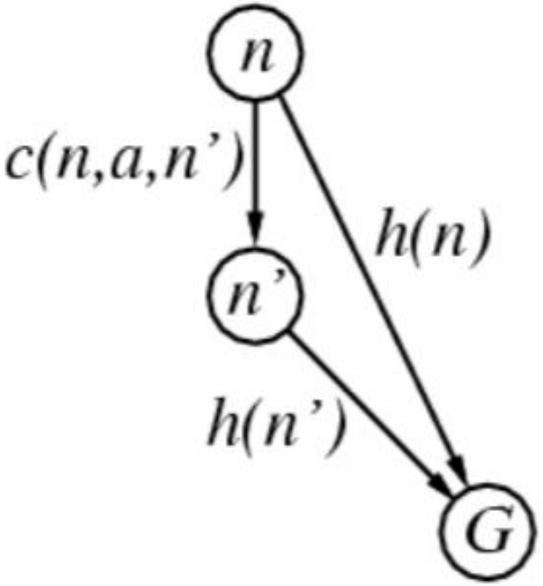
Consistent (monotone) Heuristics

- A heuristic is consistent if for every node n, every successor n' of n generated by any action a,

  $h(n) \le c(n,a,n') + h(n')$

- If h is consistent, we have

$$\begin{aligned} f(n') &= g(n') + h(n') \\ &= g(n) + c(n,a,n') + h(n') \\ &\ge g(n) + h(n) \\ &= f(n) \end{aligned}$$



- i.e., $f(n)$ is non-decreasing along any path.

- Theorem: If $h(n)$ is consistent, f along any path is non-decreasing.
- Corollary: the f values seen by A* are non-decreasing.

Sample Heuristic functions

h(n)=Number of tiles out of position.

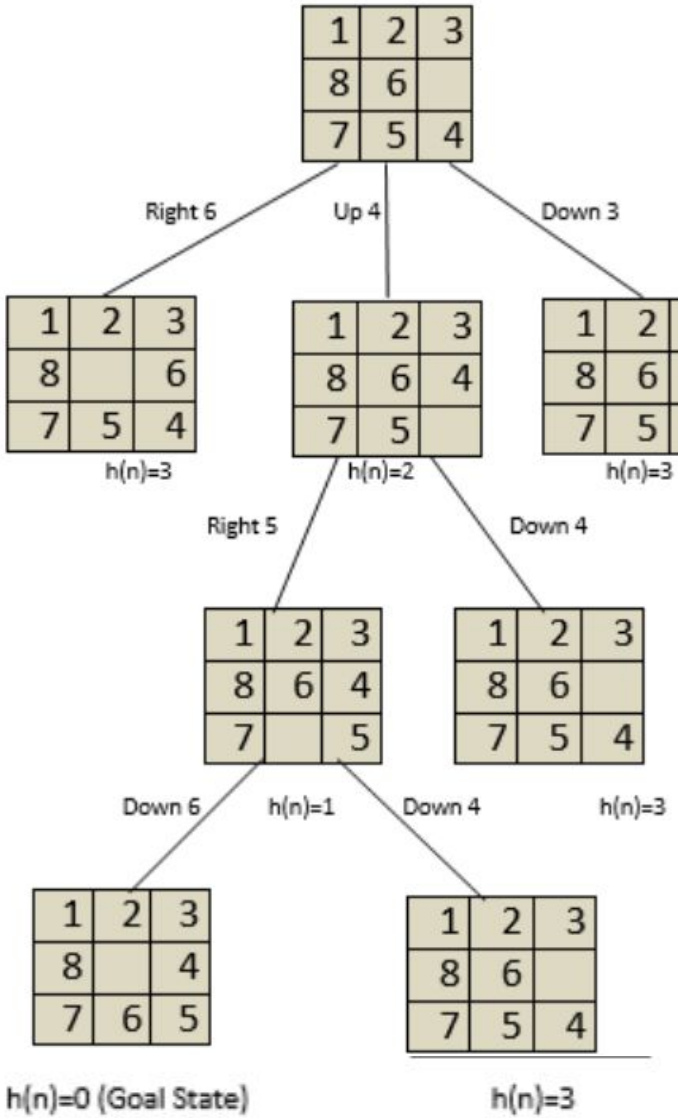
h(n)=3

1	2	3
8	6	
7	5	4

Start State

1	2	3
8		4
7	6	5

Goal State



1	2	3
	4	6
7	**5**	**8**

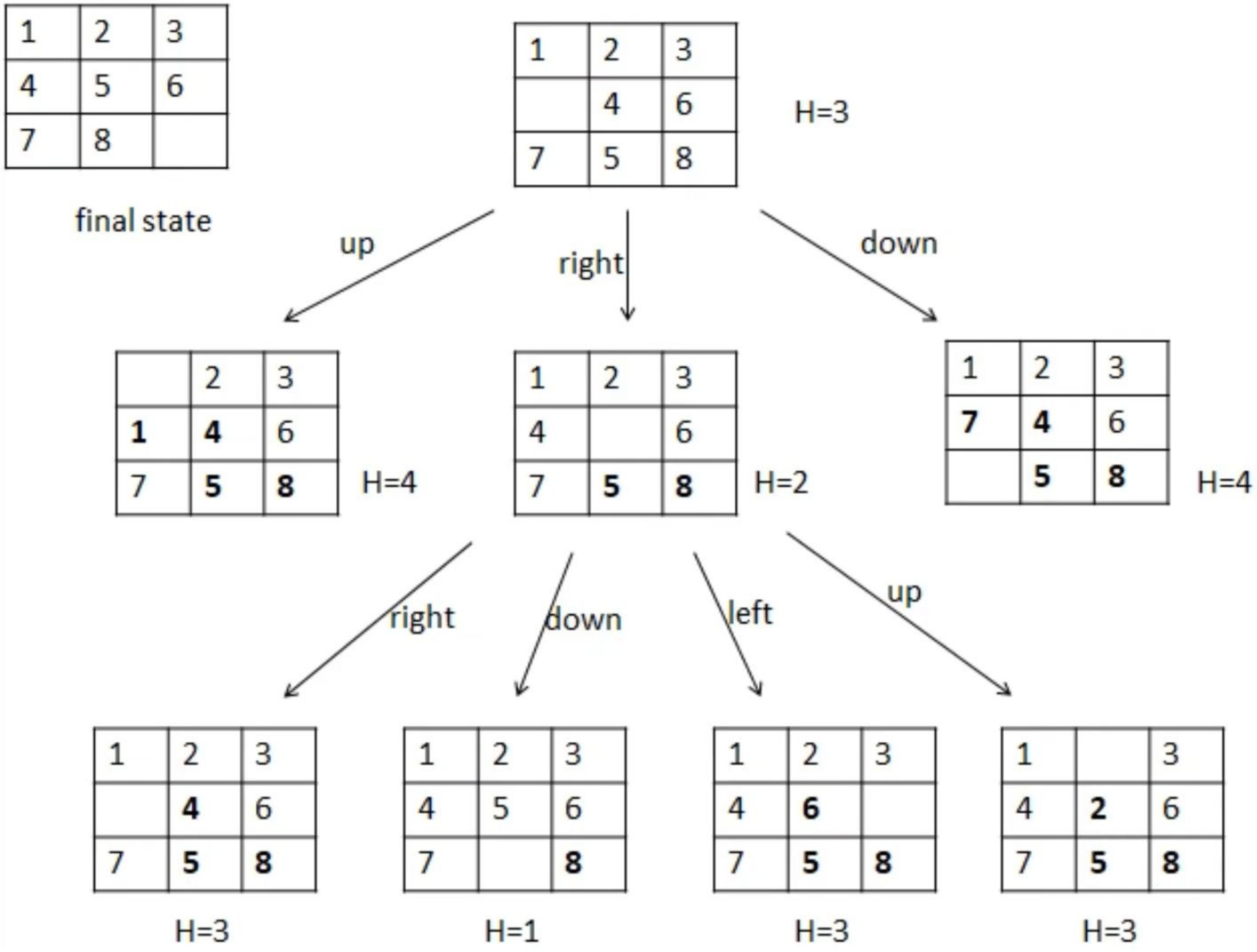
Initial state

1	2	3
4	5	6
7	8	

final state

Heuristic value=1+1+1=3

Set of action={up, down, left, right}


1 2 3
4 5 6
7 8
final state
1 2 3
4 6
7 5 8
H=3
up
right
down
2 3
1 4 6
7 5 8
H=4
1 2 3
4 6
7 5 8
H=2
1 2 3
7 4 6
5 8
H=4
right
down
left
up
1 2 3
4 6
7 5 8
H=3
1 2 3
4 5 6
7 8
H=1
1 2 3
4 6
7 5 8
H=3
1 3
4 2 6
7 5 8
H=3

7	2	4
5	*	6
8	3	1

Initial State

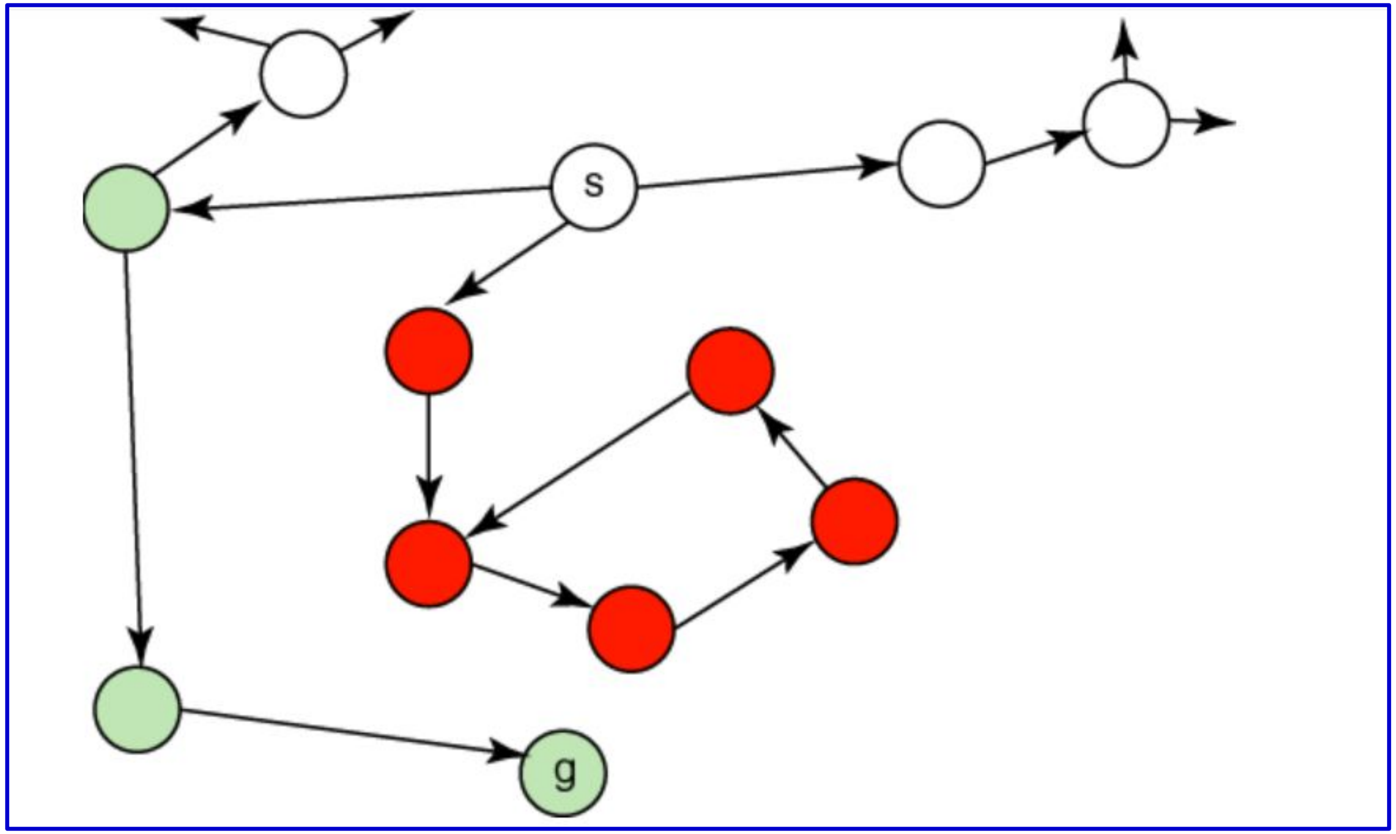
*	1	2
3	4	5
6	7	8

Final State

H1 : Number of misplaced tiles

H2: Sum of Eucledian distances of the tiles from their goal positions

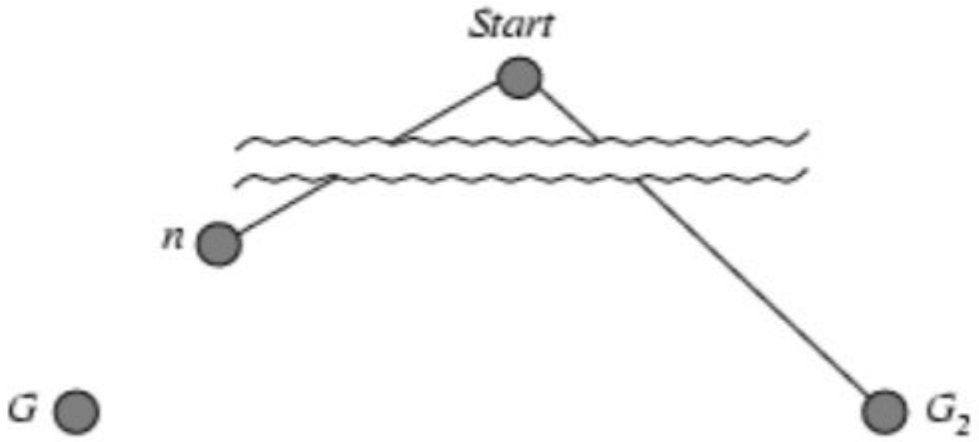
H3 : Sum of Manhattan distances of the tiles from their goal positions



A graph that is bad for best-first search

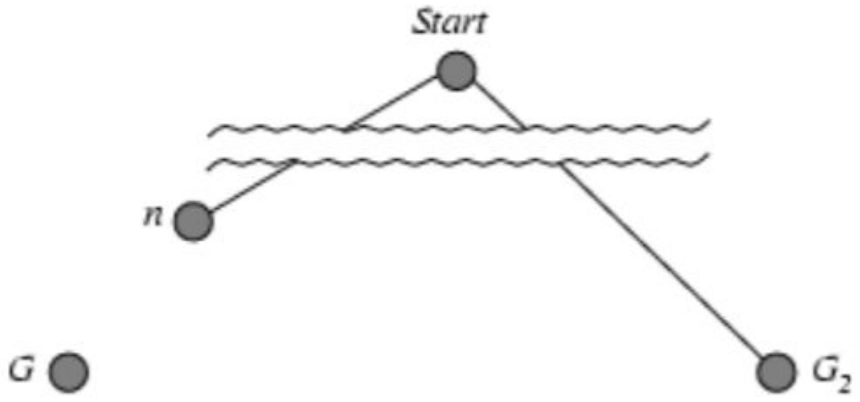
Properties of A*

Suppose some suboptimal goal G_2 has been generated and is in the fringe. Let *n* be an unexpanded node in the fringe such that *n* is on a shortest path to an optimal goal *G*.



$f(G_2)$	$= g(G_2)$		since $h(G_2) = 0$
		$> g(G)$	since G_2 is suboptimal
$f(G)$	$= g(G)$		since $h(G) = 0$
$f(G_2)$	$> f(G)$		from above

Suppose some suboptimal goal G_2 has been generated and is in the fringe. Let n be an unexpanded node in the fringe such that n is on a shortest path to an optimal goal G.



$f(G_2) > f(G)$ from prev slide

$h(n) \leq h^*(n)$ since h is admissible

$g(n) + h(n) \leq g(n) + h^*(n)$

$f(n) \leq f(G) < f(G_2)$

Hence $f(n) < f(G_2) \Rightarrow A^*$ will never select G_2 for expansion.

Properties of A*

- Complete?
- Yes, (unless there are infinitely many nodes with f<=f(G)
- Time? Exponential (Worst case all the nodes are added)
- Space ? Keeps all nodes in memory
- Optimal?

In general, A* not practical for large scale problems due to memory requirements (all generated nodes in memory)

Idea: Use iterative deepening