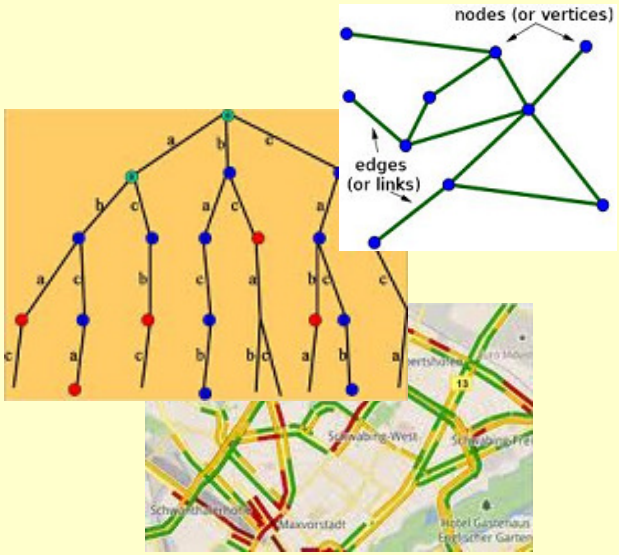


State Search Algorithms

Aliridho Barakbah

Knowledge Engineering Laboratory
Department of Information and Computer Engineering
Politeknik Elektronika Negeri Surabaya

Deskripsi

- Merupakan algoritma untuk mencari kemungkinan penyelesaian
- Sering dijumpai oleh peneliti di bidang AI

Mendefinisikan permasalahan

- Mendefinisikan suatu state (ruang keadaan)
- Menerapkan satu atau lebih state awal
- Menetapkan satu atau lebih state tujuan
- Menetapkan rules (kumpulan aturan)

Contoh kasus

Seorang petani ingin memindah dirinya sendiri, seekor serigala, seekor angsa gemuk, dan seikat padi yang berisi menyeberangi sungai. Sayangnya, perahunya sangat terbatas; dia hanya dapat membawa satu objek dalam satu penyeberangan. Dan lagi, dia tidak bisa meninggalkan serigala dan angsa dalam satu tempat, karena serigala akan memangsa angsa. Demikian pula dia tidak bisa meninggalkan angsa dengan padi dalam satu tempat.

State (ruang keadaan)

- State → (Serigala, Angsa, Padi, Petani)
- Daerah asal ketika hanya ada serigala dan padi, dapat direpresentasikan dengan state (1, 0, 1, 0), sedangkan daerah tujuan adalah (0, 1, 0, 1)

State awal dan tujuan

- State awal
  - Daerah asal → (1, 1, 1, 1)
  - Daerah tujuan → (0, 0, 0, 0)
- State tujuan
  - Daerah asal → (0, 0, 0, 0)
  - Daerah tujuan → (1, 1, 1, 1)

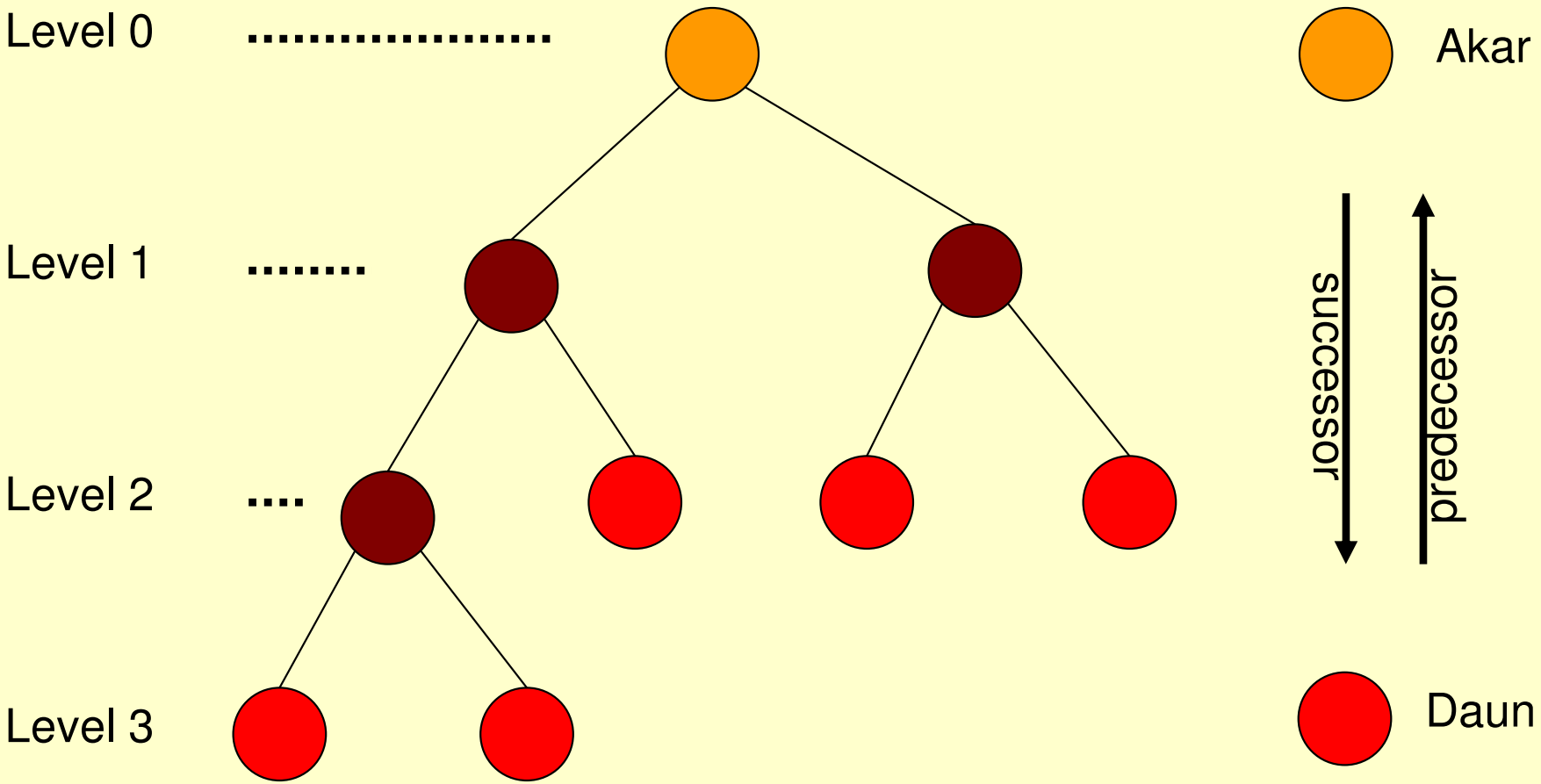
Rules

Aturan ke	Rule
1	Angsa menyeberang bersama petani
2	Padi menyeberang bersama petani
3	Serigala menyeberang bersama petani
4	Angsa kembali bersama petani
5	Padi kembali bersama petani
6	Serigala kembali bersama petani
7	Petani kembali

Contoh solusi

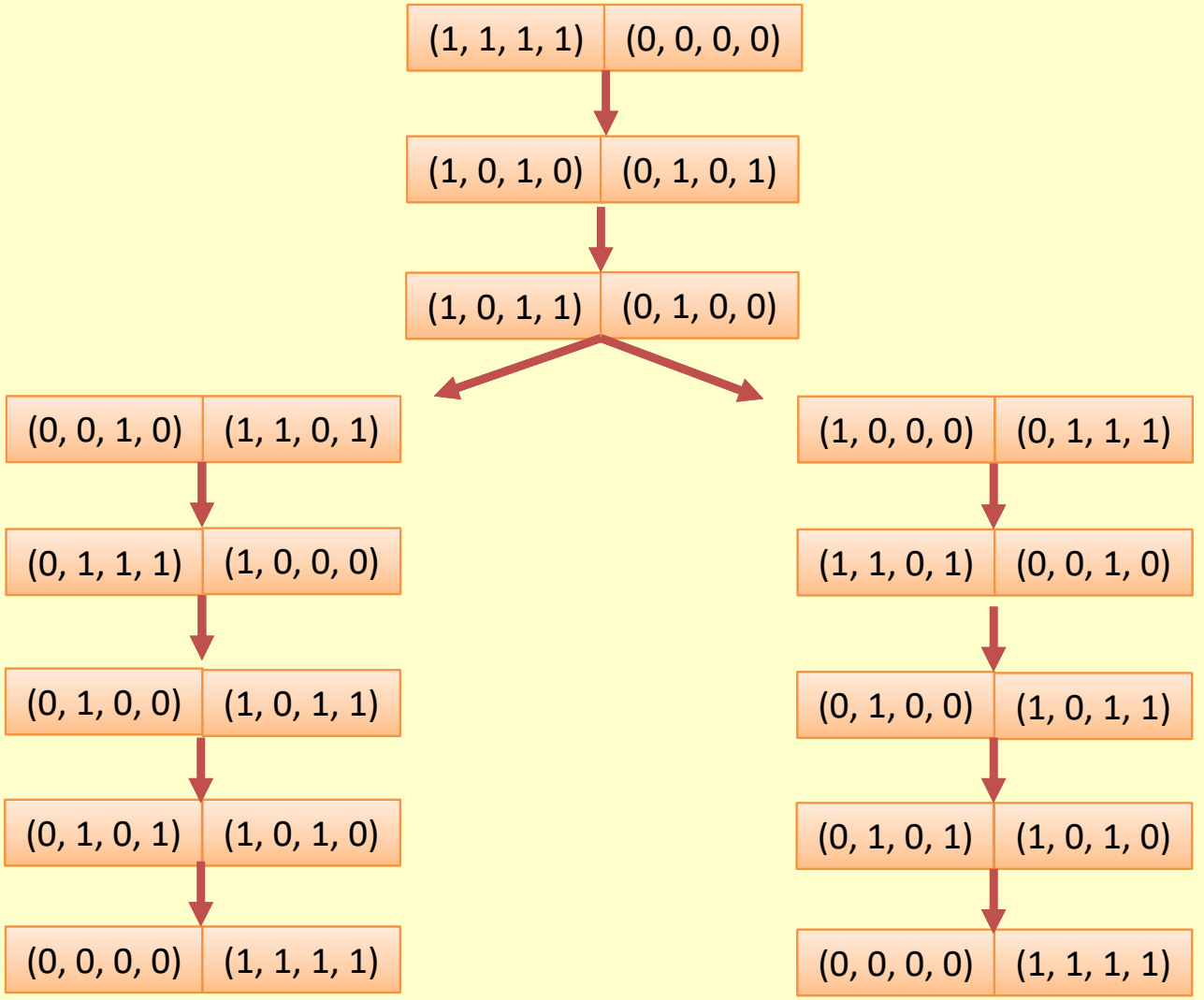
Daerah asal (S, A, Pd, Pt)	Daerah tujuan (S, A, Pd, Pt)	Rule yang dipakai
(1, 1, 1, 1)	(0, 0, 0, 0)	1
(1, 0, 1, 0)	(0, 1, 0, 1)	7
(1, 0, 1, 1)	(0, 1, 0, 0)	3
(0, 0, 1, 0)	(1, 1, 0, 1)	4
(0, 1, 1, 1)	(1, 0, 0, 0)	2
(0, 1, 0, 0)	(1, 0, 1, 1)	7
(0, 1, 0, 1)	(1, 0, 1, 0)	1
(0, 0, 0, 0)	(1, 1, 1, 1)	solusi

Pohon pelacakan

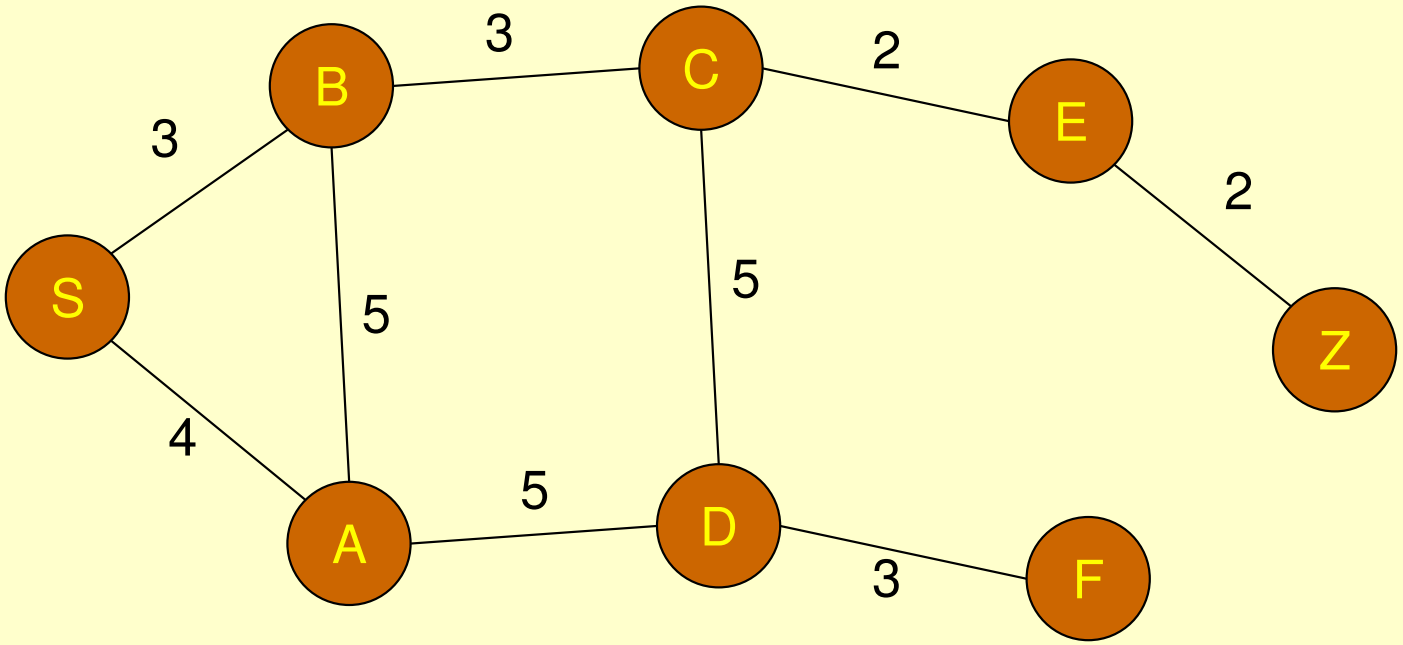


Representasi State

Daerah Asal	Daerah Tujuan
(Serigala, Angsa, Padi, Petani)	(Serigala, Angsa, Padi, Petani)

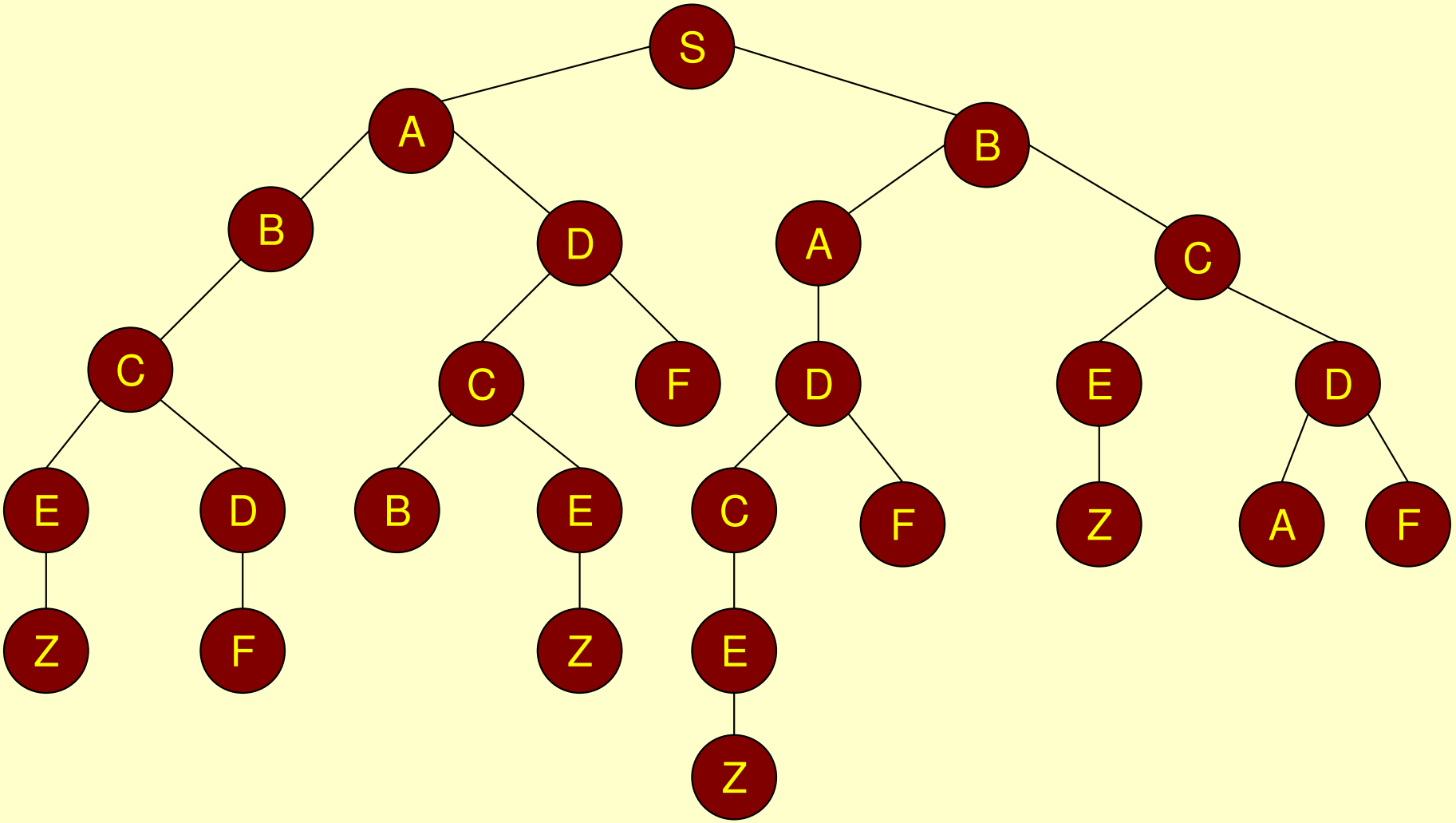


Contoh kasus

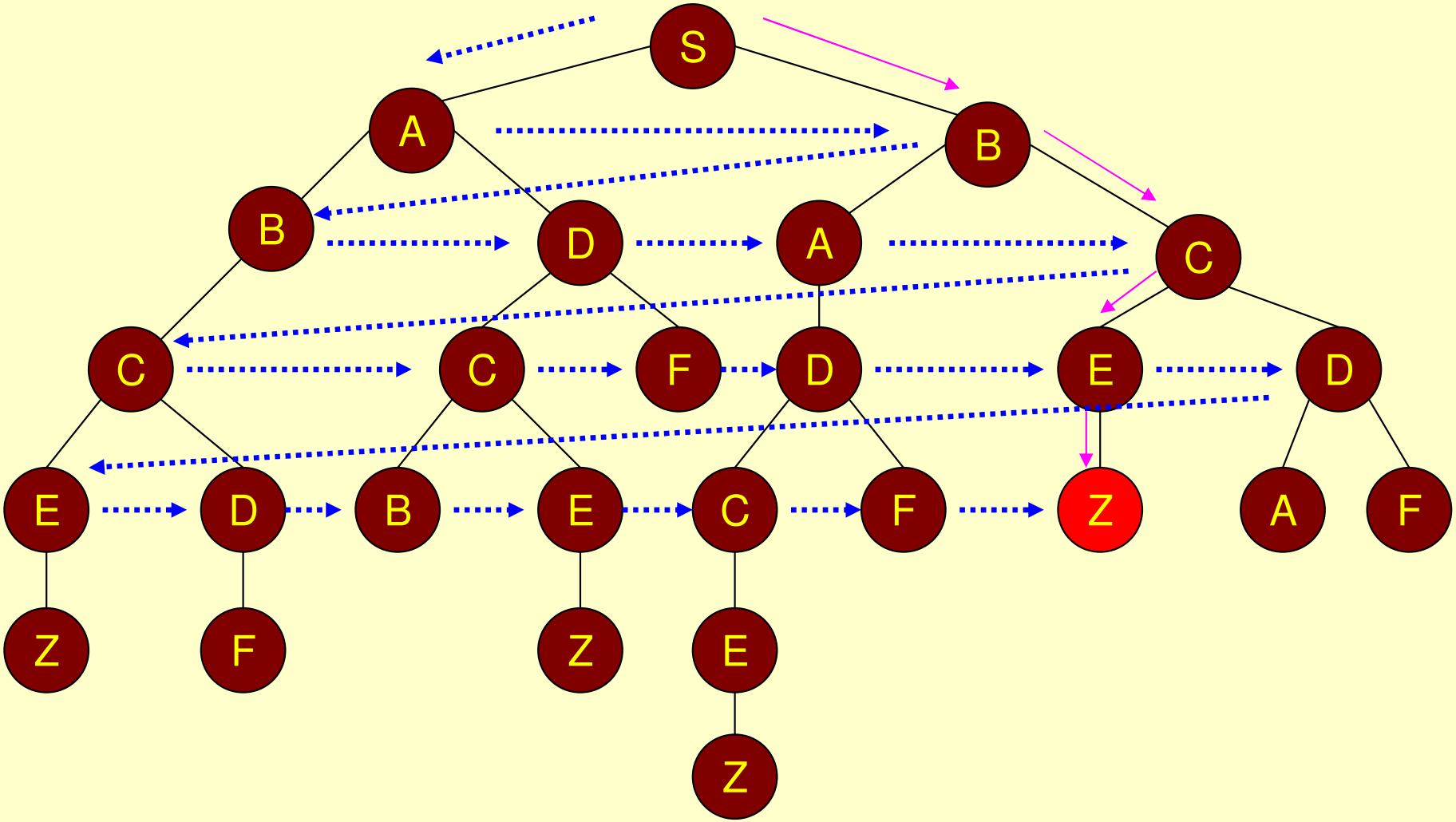


Graph yang memuat Hamiltonian circuit

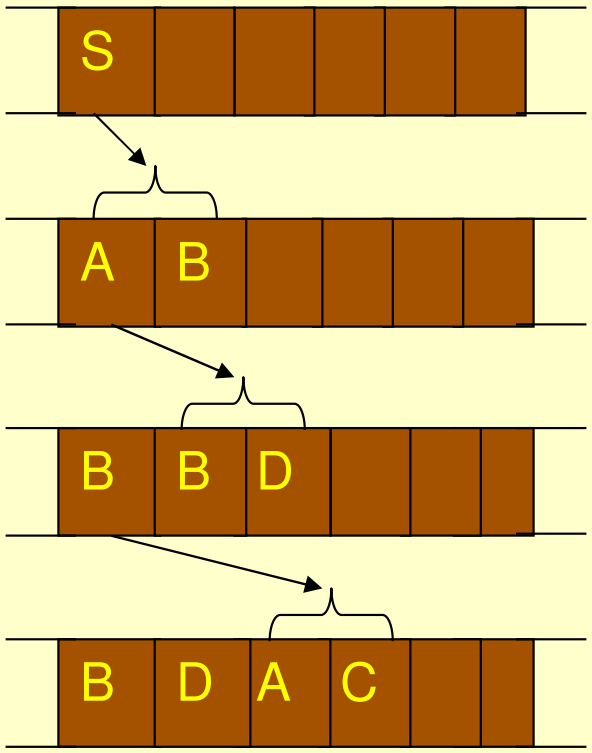
Susunan pohon



Breadth First Search



Algoritma

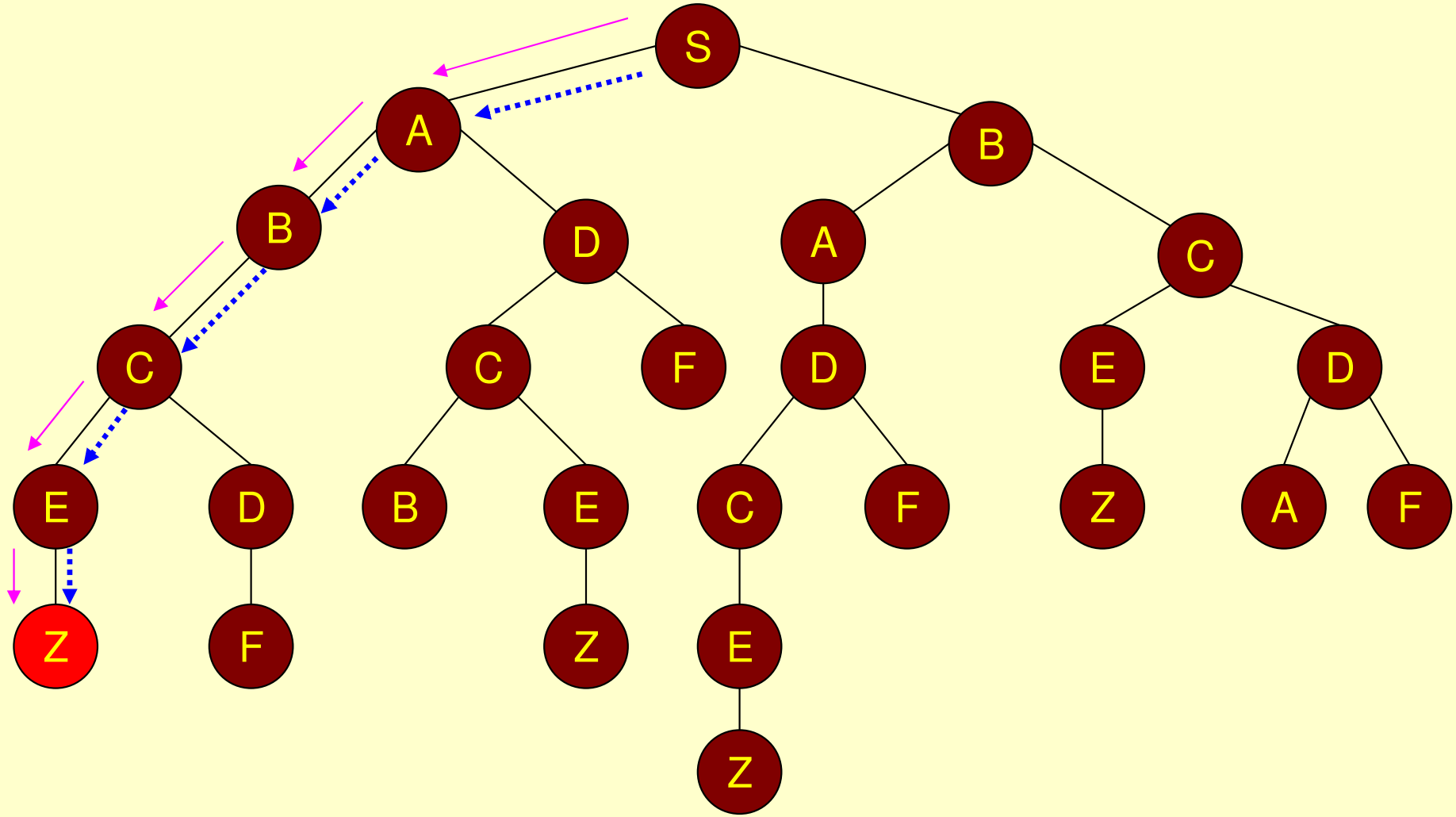


dan seterusnya …

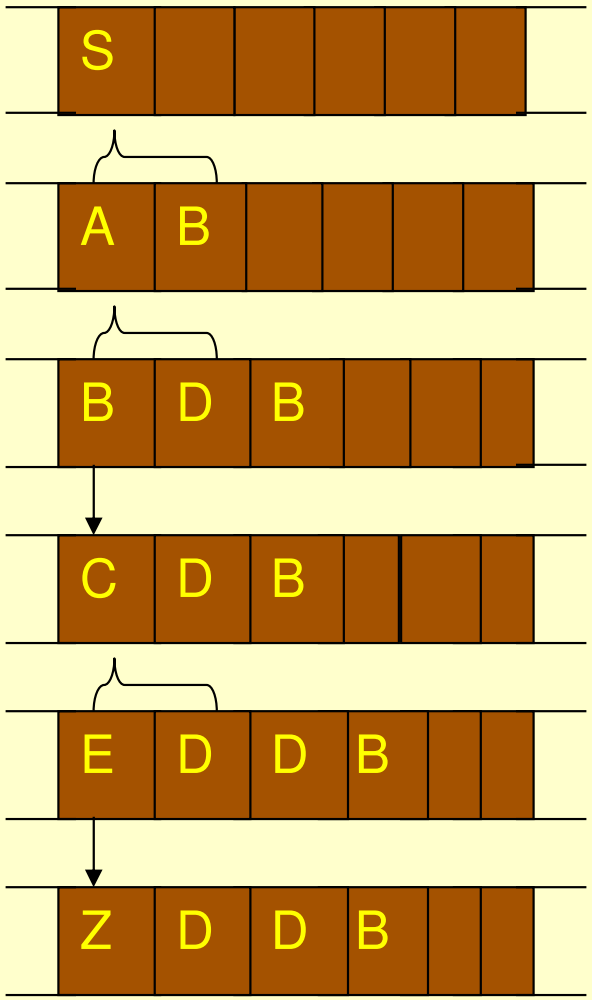
Analisa

- Kelebihan
  - Tidak akan menemui jalan buntu
  - Jika ada satu solusi, pasti diketemukan
- Kelemahan
  - Boros memori
  - Mungkin terjebak pada local optima

Depth First Search



Algoritma



Analisa

- Kelebihan
  - Butuh memori yang relatif kecil
  - Menemukan solusi tanpa harus menguji lebih banyak lagi
- Kelemahan
  - Mungkin terjebak pada local optima

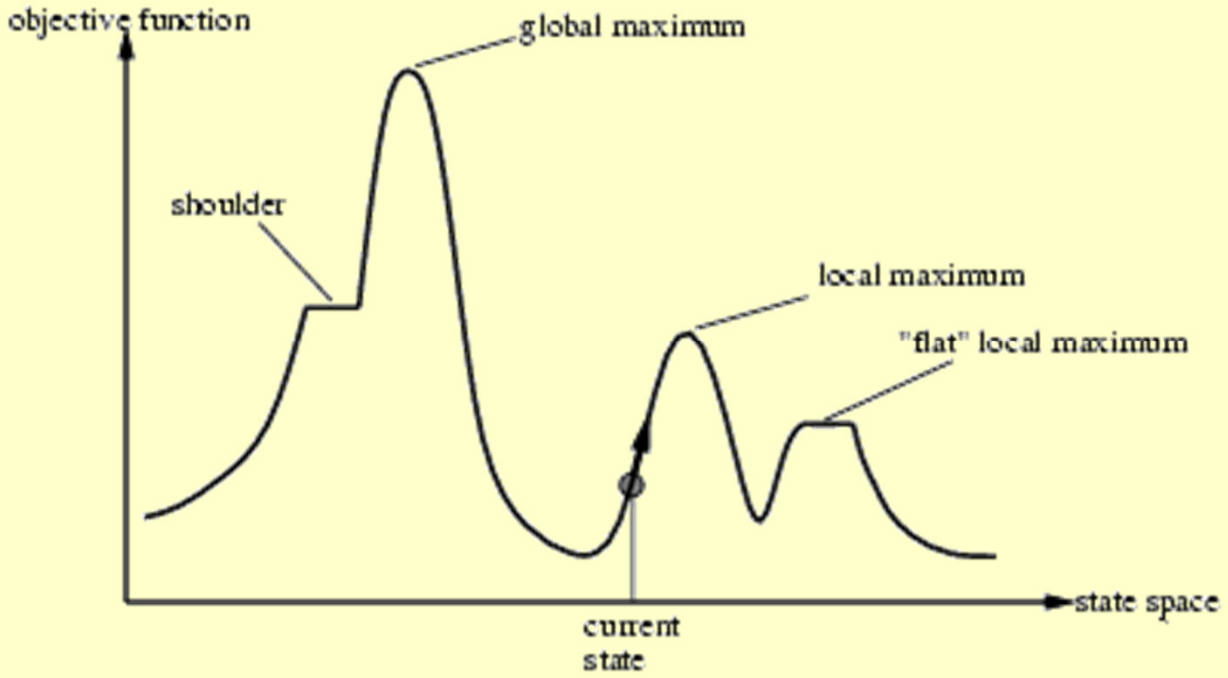
Hill Climbing Search

- "Like climbing Everest in thick fog with amnesia"

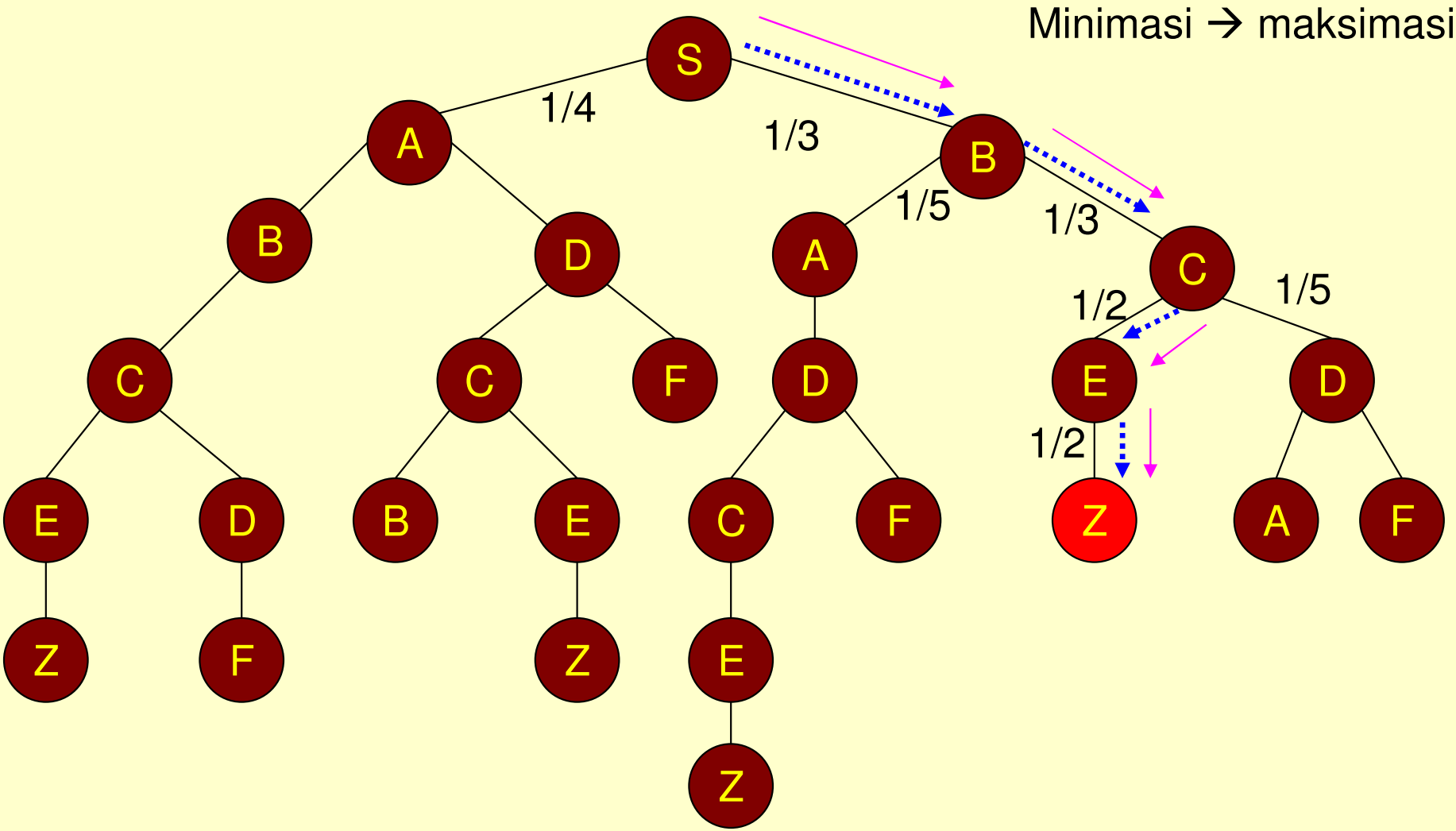
```
function HILL-CLIMBING( problem) returns a state that is a local maximum
    inputs: problem, a problem
    local variables: current, a node
                     neighbor, a node

    current ← MAKE-NODE(INITIAL-STATE[problem])
    loop do
        neighbor ← a highest-valued successor of current
        if VALUE[neighbor] ≤ VALUE[current] then return STATE[current]
        current ← neighbor
```

Hill-climbing search

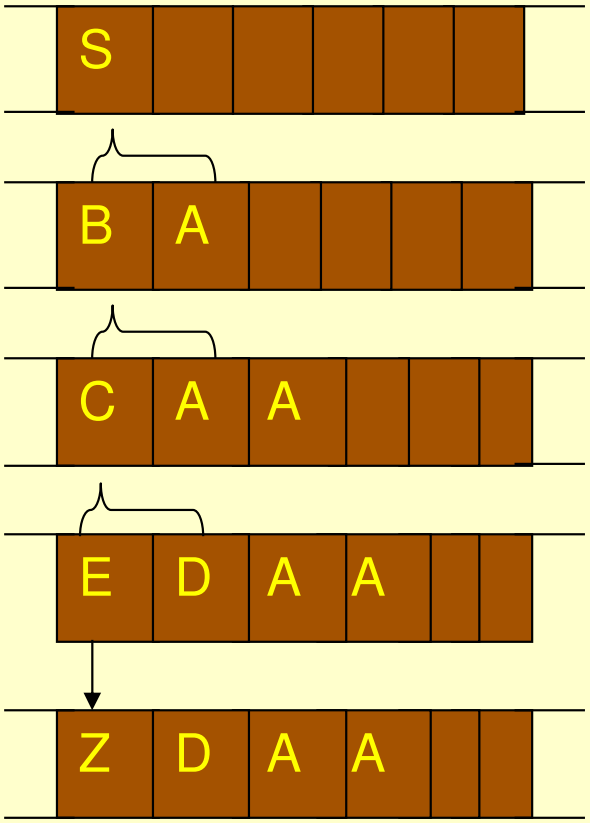
- Problem: depending on initial state, can get stuck in local maxima



Hill climbing



Algoritma



Analisa

- Kelebihan
  - Butuh memori kecil
  - Menemukan solusi tanpa harus menguji lebih banyak lagi
- Kelemahan
  - Mungkin terjebak pada local optima

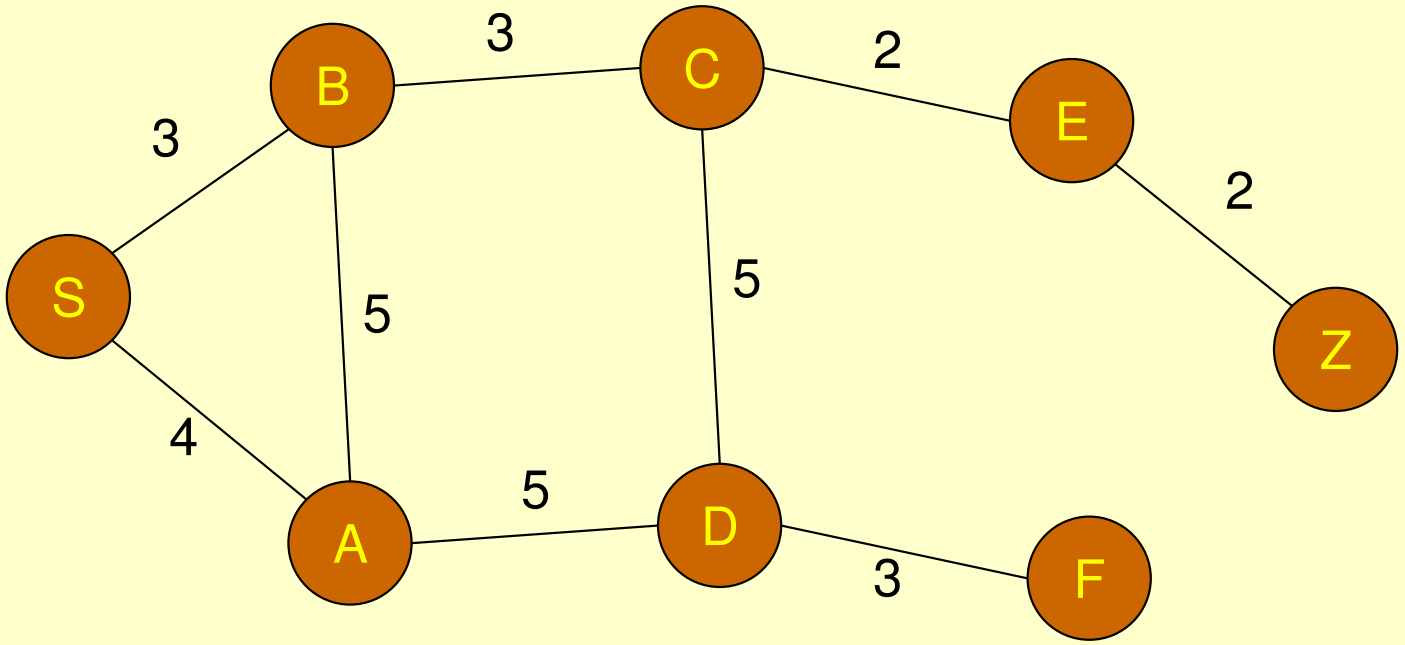
Best First Search

Termasuk Greedy Search

Heuristic Function

f(n) = h(n)

where h(n) = estimated cost to goal from n



Perkiraan Jarak

Node	Perkiraan Jarak ke Z
S	10
A	8
B	6
C	3,7
D	4,5
E	2
F	2

Heuristic

Webster's Revised Unabridged Dictionary (1913) (web1913)

Heuristic \Heu*ris"tic\, a. [Greek. to discover.] Serving to discover or find out.

The Free On-line Dictionary of Computing (15Feb98)

heuristic 1. <programming> A rule of thumb, simplification or educated guess that reduces or limits the search for solutions in domains that are difficult and poorly understood. Unlike algorithms, heuristics do not guarantee feasible solutions and are often used with no theoretical guarantee. 2. <algorithm> approximation algorithm.

From WordNet (r) 1.6

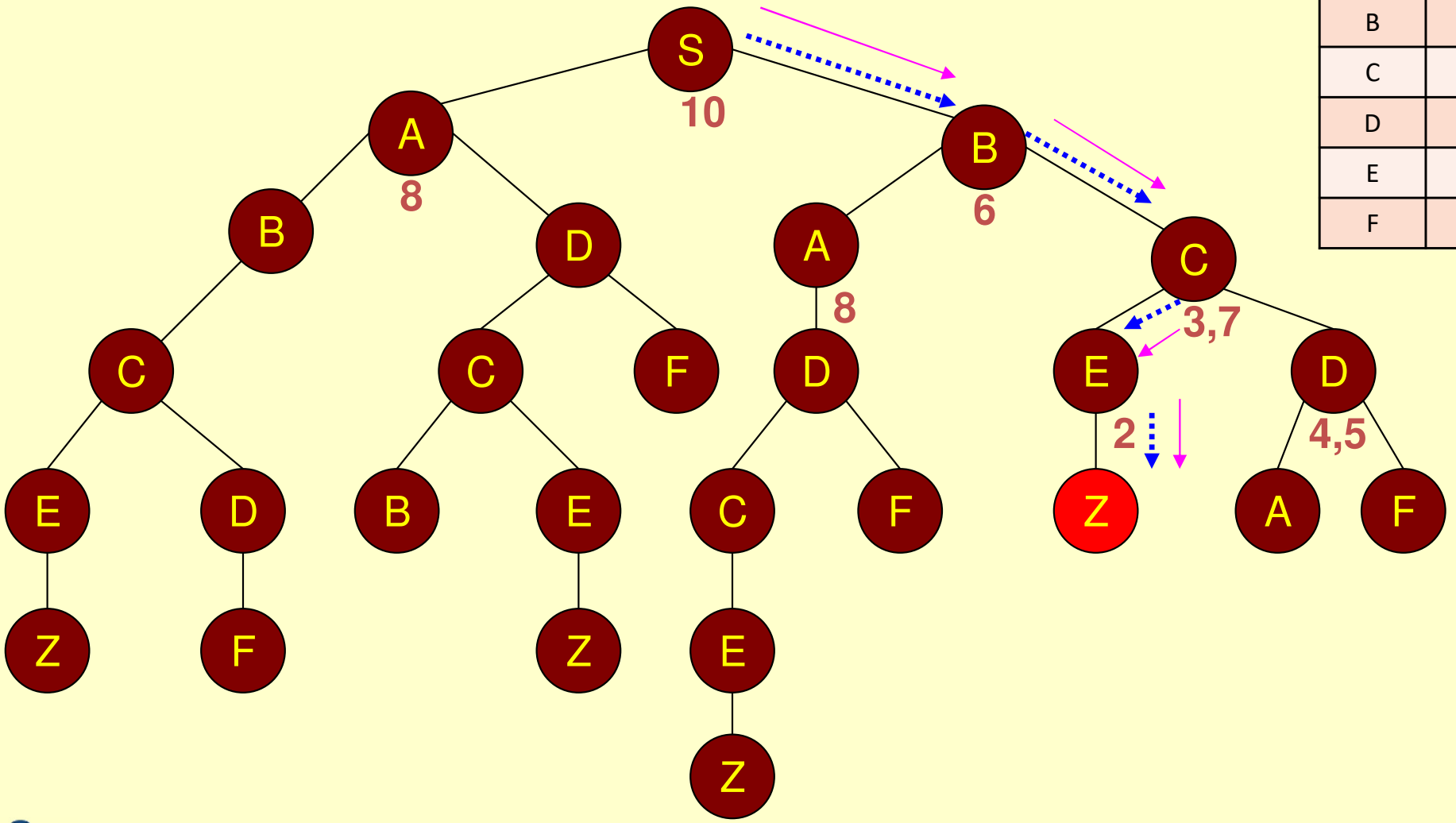
heuristic adj 1: (computer science) relating to or using a heuristic rule 2: of or relating to a general formulation that serves to guide investigation [ant: algorithmic] n : a commonsense rule (or set of rules) intended to increase the probability of solving some problem [syn: heuristic rule, heuristic program]

Heuristic Search

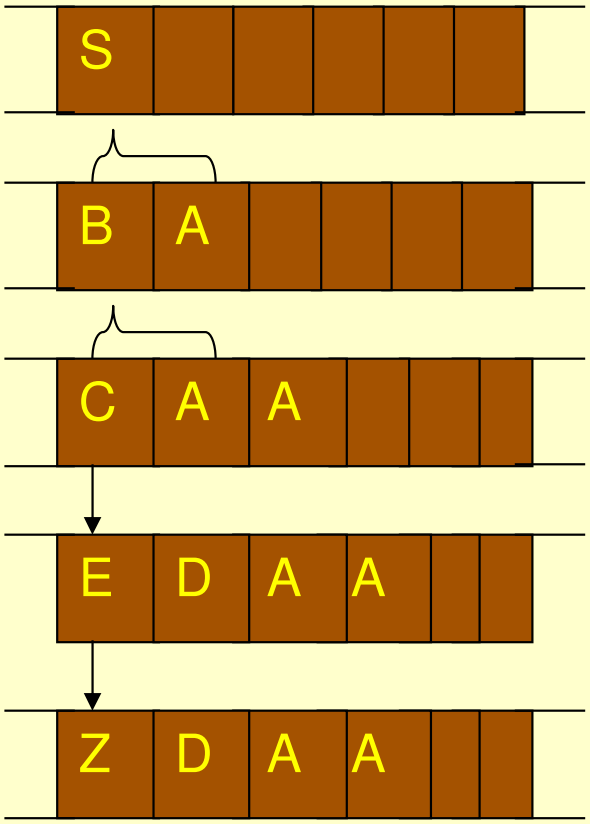
- Define a heuristic function, **h(n)**, that estimates the "goodness" of a node n.
- Specifically, h(n) = **estimated cost** (or distance) of minimal cost path from n **to a goal state**.
- The heuristic function is an estimate, based on domain-specific information that is computable from the current state description, of how close we are to a goal

Best First Search

Node	Perkiraan Jarak ke Z
S	10
A	8
B	6
C	3,7
D	4,5
E	2
F	2



Algoritma



Analisa

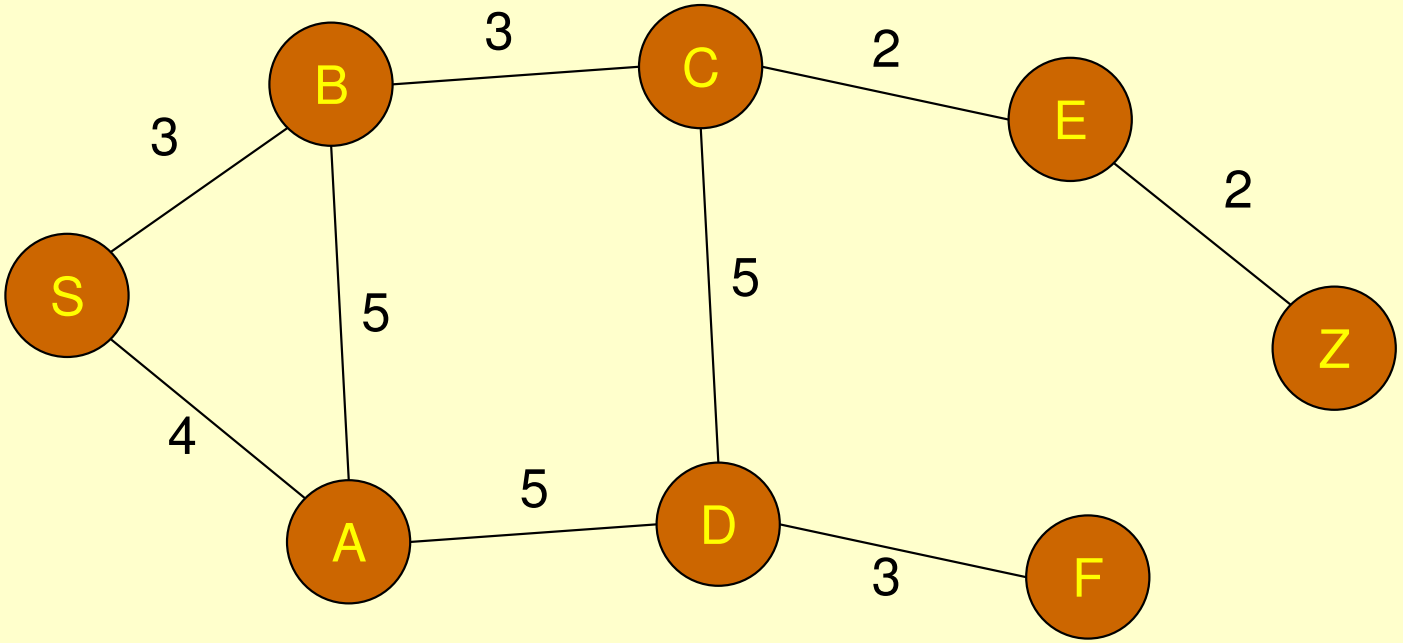
- Kelebihan
  - Butuh memori kecil
  - Menemukan solusi tanpa harus menguji lebih banyak lagi
- Kelemahan
  - Mungkin terjebak pada local optima

A* Search

Termasuk Informed Search

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

where
g(n) = cost so far to reach n
h(n) = estimated cost to goal from n



Perkiraan Jarak

Node	Perkiraan Jarak ke Z
S	10
A	8
B	6
C	3,7
D	4,5
E	2
F	2

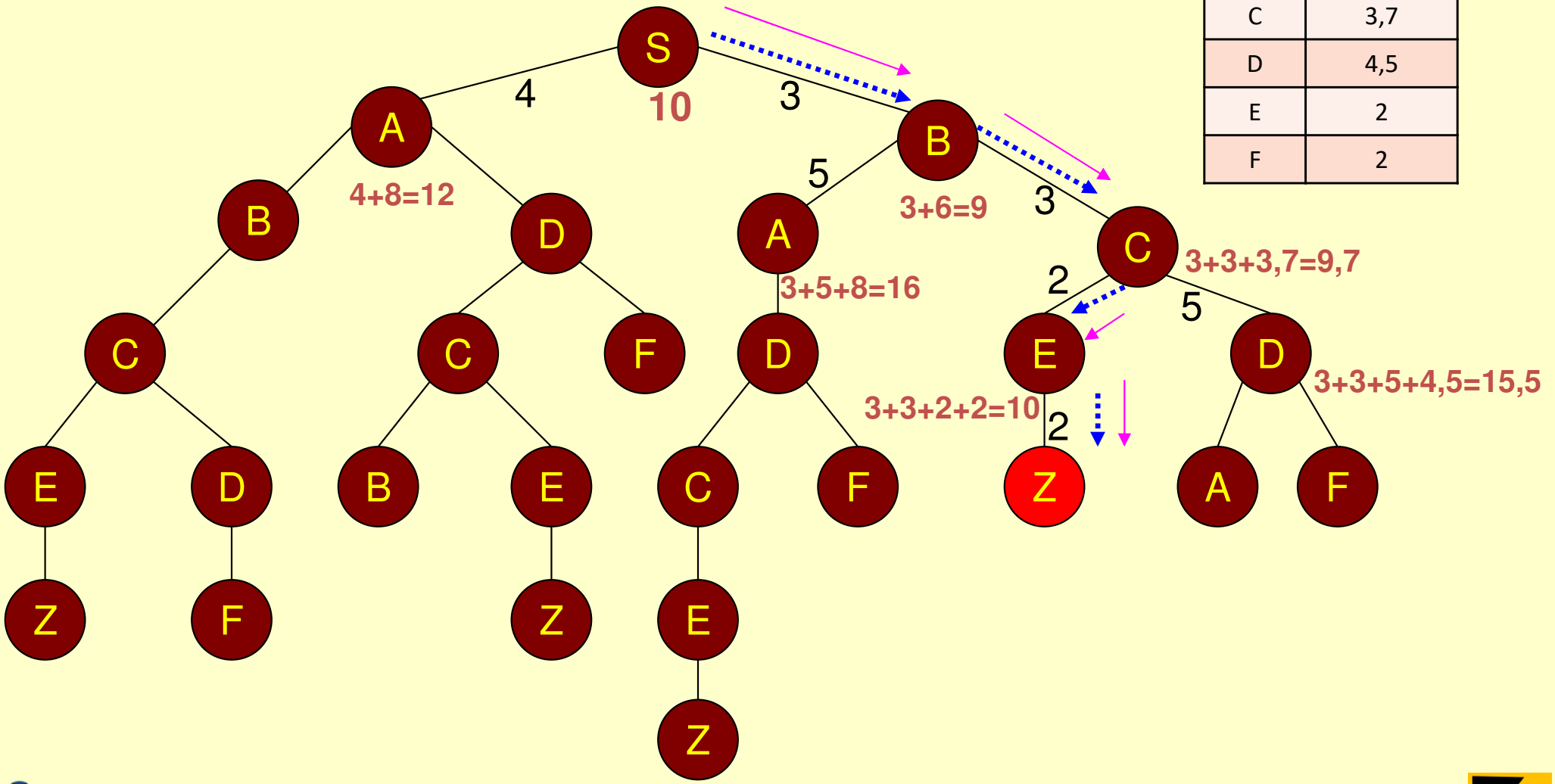
Informed Search

- Add domain-specific information to select the best path along which to continue searching
- We kept looking at nodes closer and closer to the goal, but were accumulating costs as we got further from the initial state
- Our goal is not to minimize the distance from the current head of our path to the goal, we want to minimize the *overall* length of the path to the goal!
- Let g(N) be the cost of the best path found so far between the initial node and N

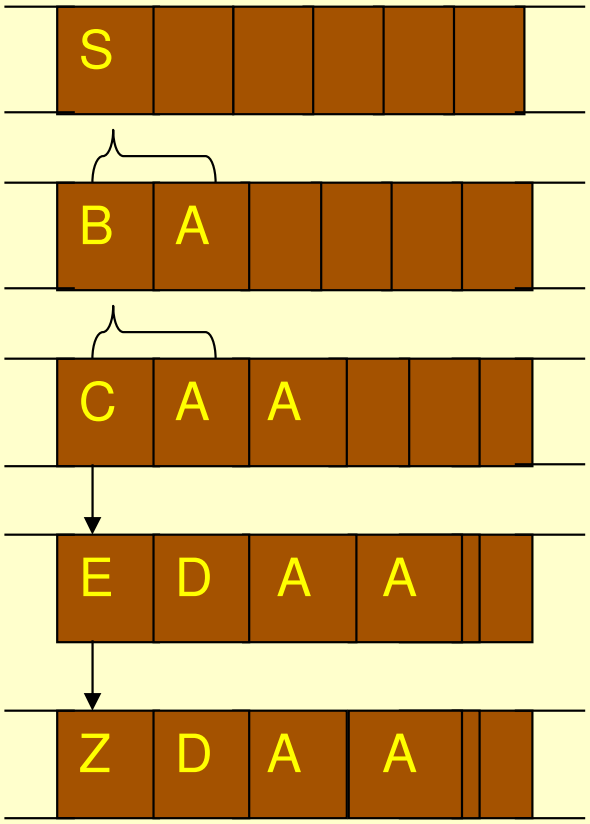
- $f(N) = g(N) + h(N)$

A* Search

Node	Perkiraan Jarak ke Z
S	10
A	8
B	6
C	3,7
D	4,5
E	2
F	2



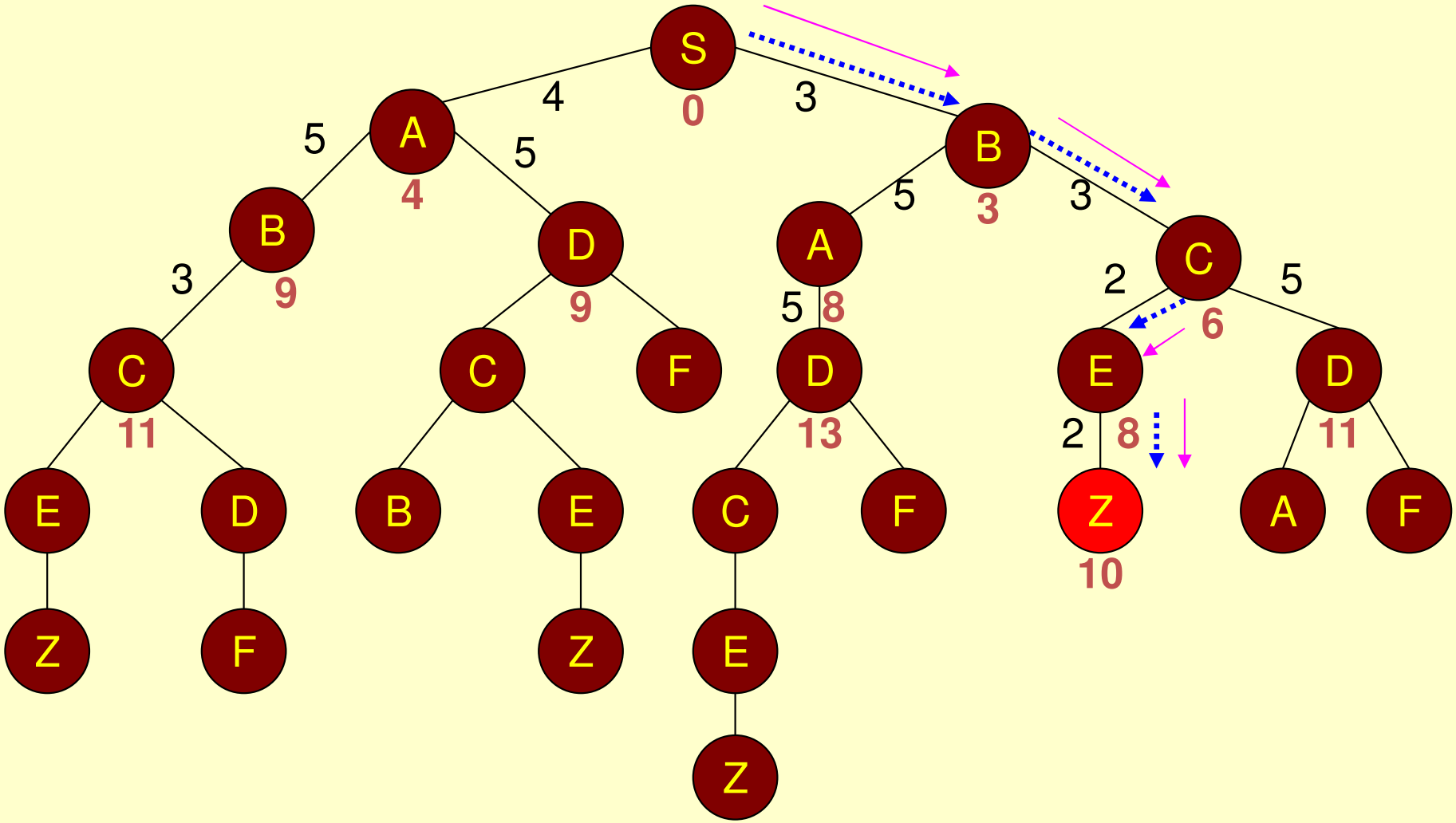
Algoritma



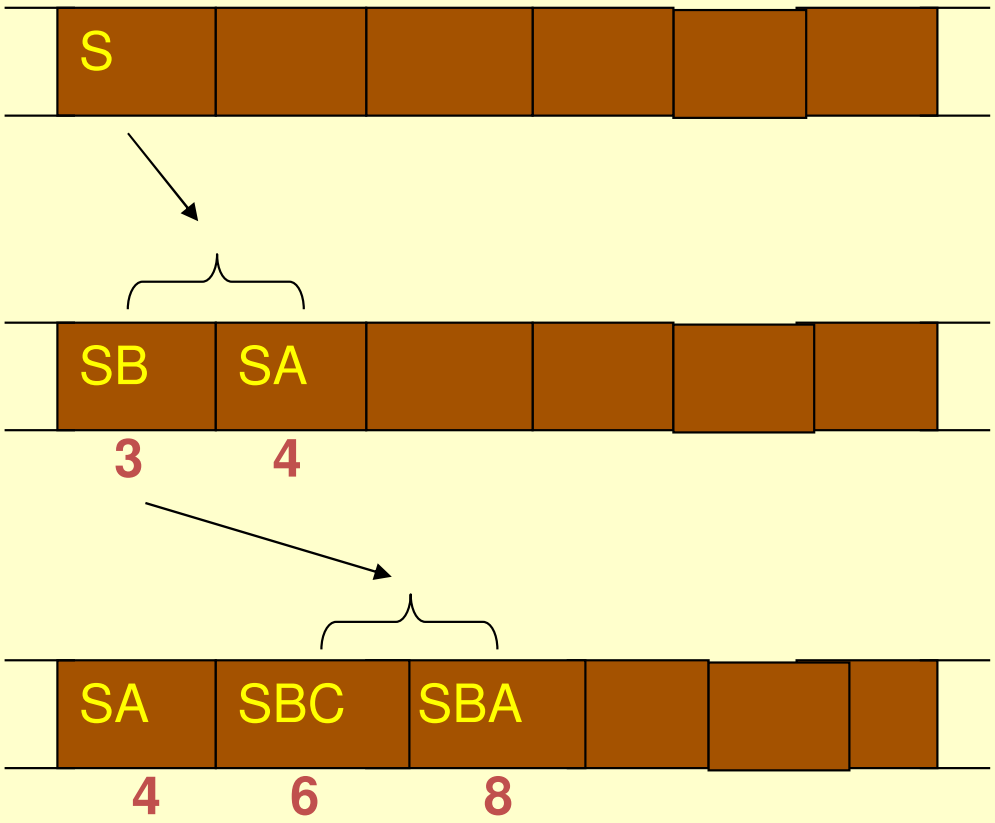
Analisa

- Kelebihan
  - Lebih baik dari Best First Search karena melibatkan factual cost
  - Butuh memori kecil
  - Menemukan solusi tanpa harus menguji lebih banyak lagi
- Kelemahan
  - Mungkin terjebak pada local optima

Branch and Bound



Algoritma

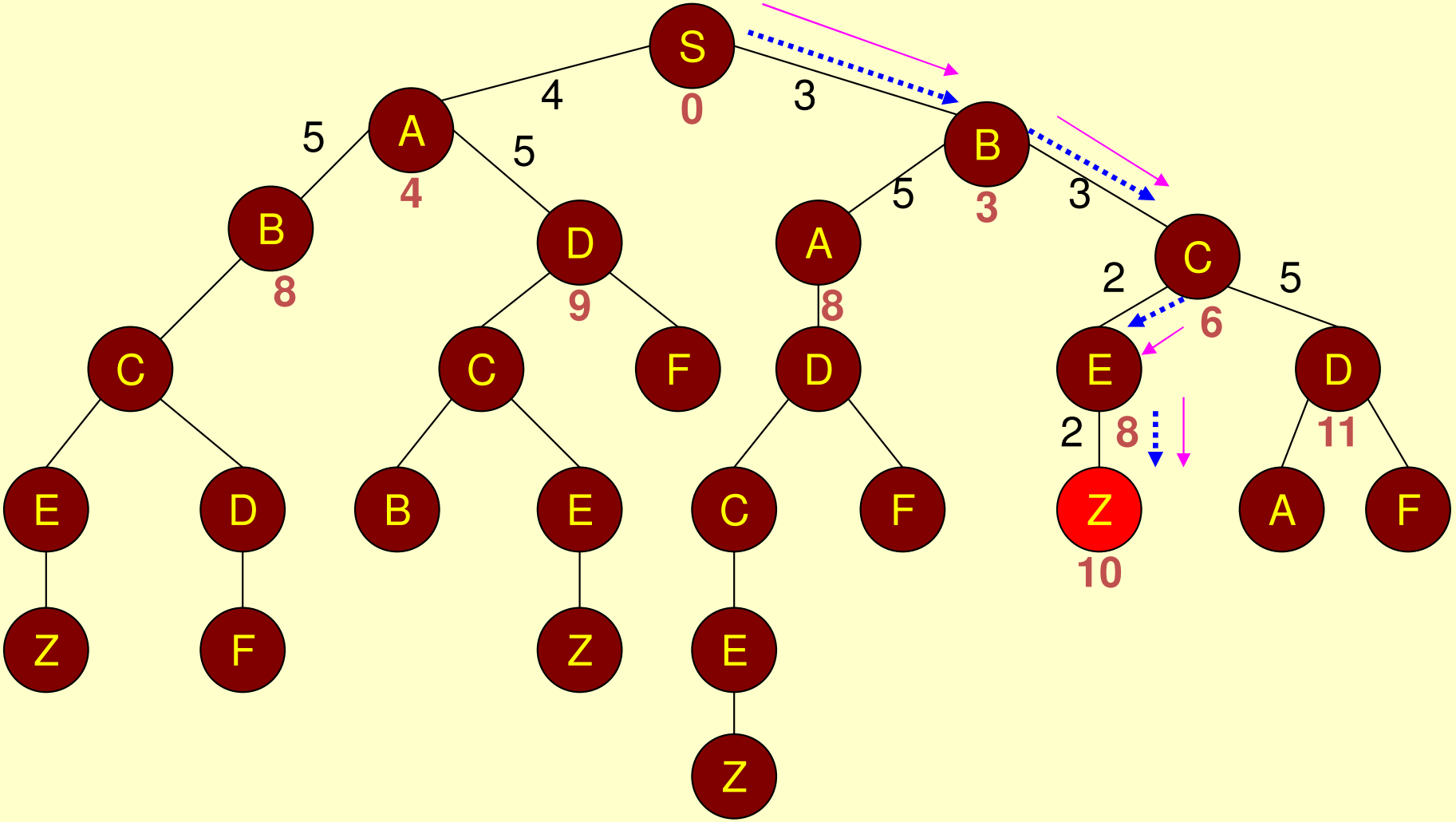


dan seterusnya…

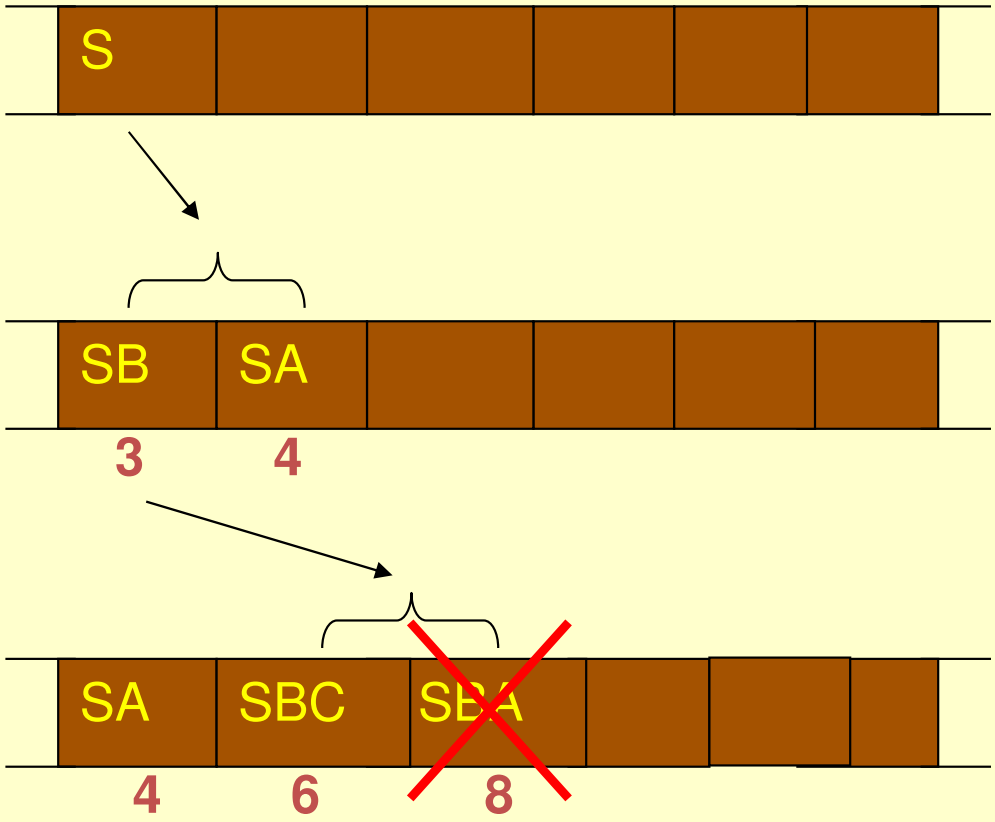
Analisa

- Kelebihan
  - Selalu menemukan global optimum
- Kelemahan
  - Boros memori karena menyimpan lintasan partial lebih dari 1 kali

Dynamic Programming



Algoritma



dan seterusnya…

Analisa

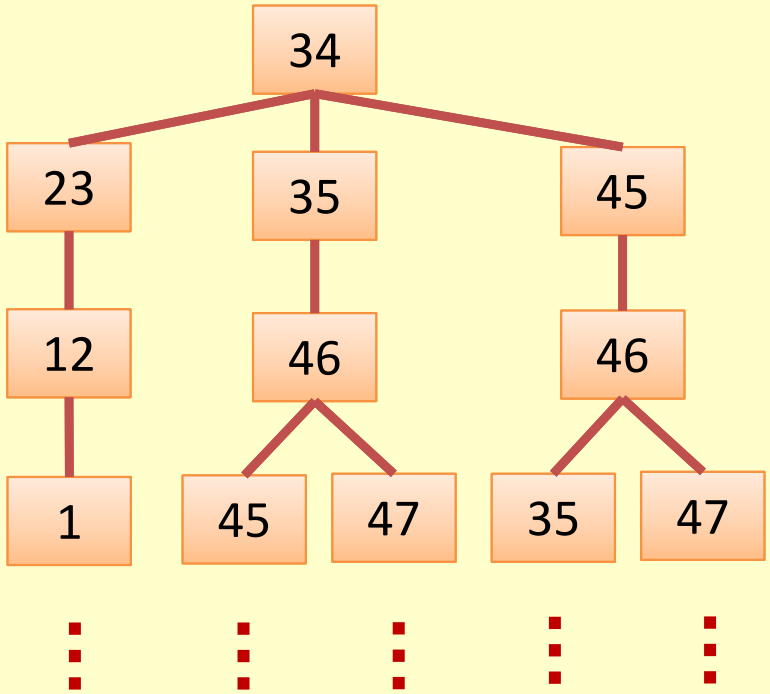
- Kelebihan
  - Selalu menemukan global optimum
  - Lebih cepat dan hemat memori karena hanya 1 kali menyimpan lintasan partial
- Kelemahan
  - Harus mengingat node terakhir dari lintasan partial yang sudah dicapai sebelumnya

Applications

- Games
- Robot Navigation
- Graph based problems
- Travelling Salesman Problem
- etc

Robot Navigation

1	2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30	31	32	33
34	35	36	37	38	39	40	41	42	43	44
45	46	47	48	49	50	51	52	53	54	55



Robot Navigation

f(N) = h(N), with h(N) = Manhattan distance to the goal

8	7	6	5	4	3	2	3	4	5	6
7		5	4	3						5
6			3	2	1	0	1	2		4
7	6									5
8	7	6	5	4	3	2	3	4	5	6

Robot Navigation

Best First Search - Heuristic Search

8	7	6	5	4	3	2	3	4	5	6
7		5	4	3						5
6			3	2	1	0	1	2		4
7	6									5
8	7	6	5	4	3	2	3	4	5	6

f(N) = h(N), with h(N) = Manhattan distance to the goal

Robot Navigation

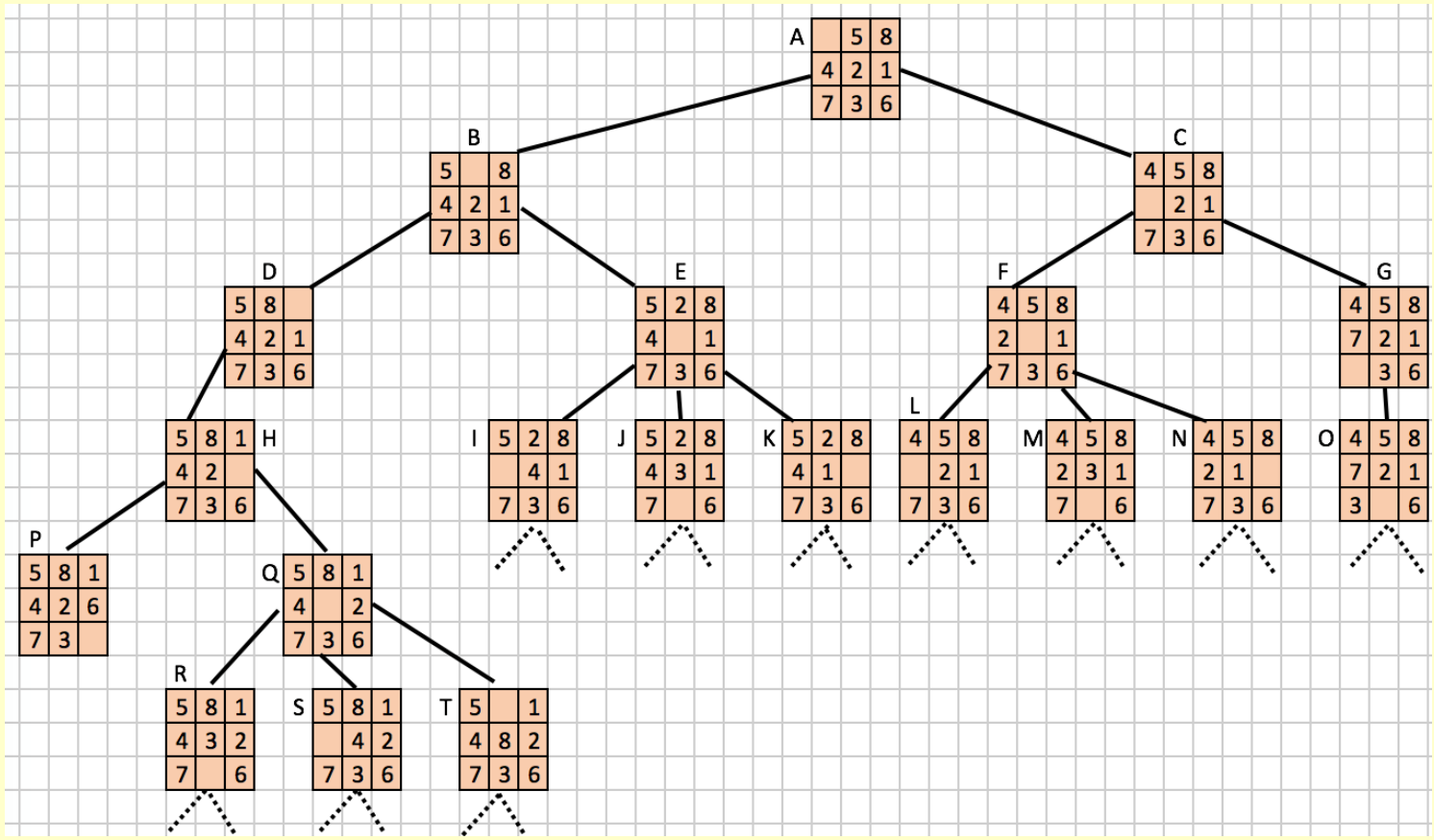
A* - Informed Search

8+3	7+4	6+3	5+6	4+7	3+8	2+9	3+10	4	5	6
7+2		5+6	4+7	3+8						5
6+1			3	2+9	1+10	0+11	1	2		4
7+0	6+1									5
8+1	7+2	6+3	5+4	4+5	3+6	2+7	3+8	4	5	6

f(N) = g(N)+h(N), with h(N) = Manhattan distance to goal

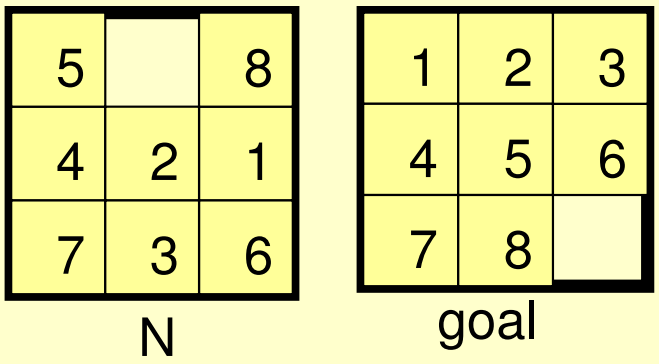
Puzzle Game

	5	8
4	2	1
7	3	6



Puzzle Game

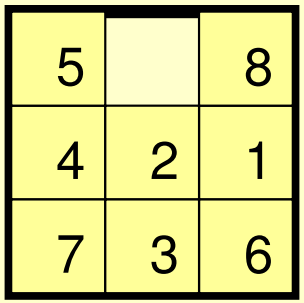
- Function h(N) that estimates the cost of the cheapest path from node N to goal node.
- Example: 8-puzzle



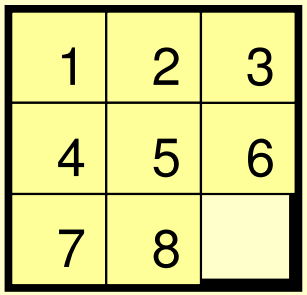
h(N) = number of misplaced tiles
= 6

Heuristic Function

- Function h(N) that estimate the cost of the cheapest path from node N to goal node.
- Example: 8-puzzle



N



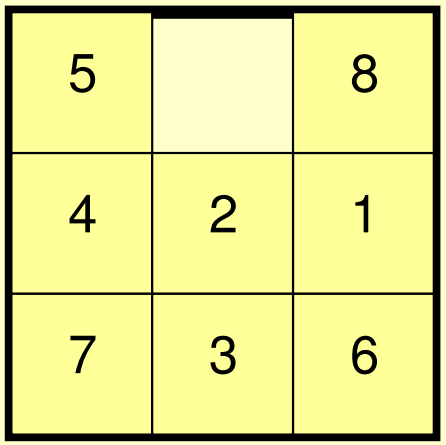
goal

h(N) = sum of the distances of every tile to its goal position

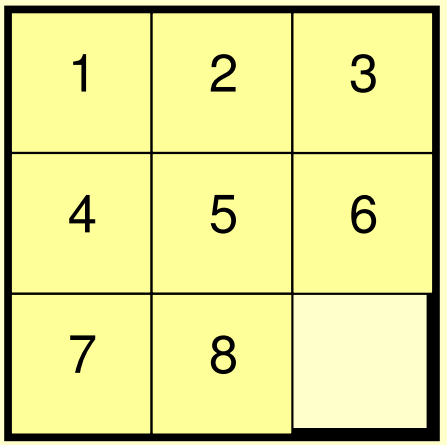
= 2 + 3 + 0 + 1 + 3 + 0 + 3 + 1

= 13

Heuristic Function for 8-Puzzle



N

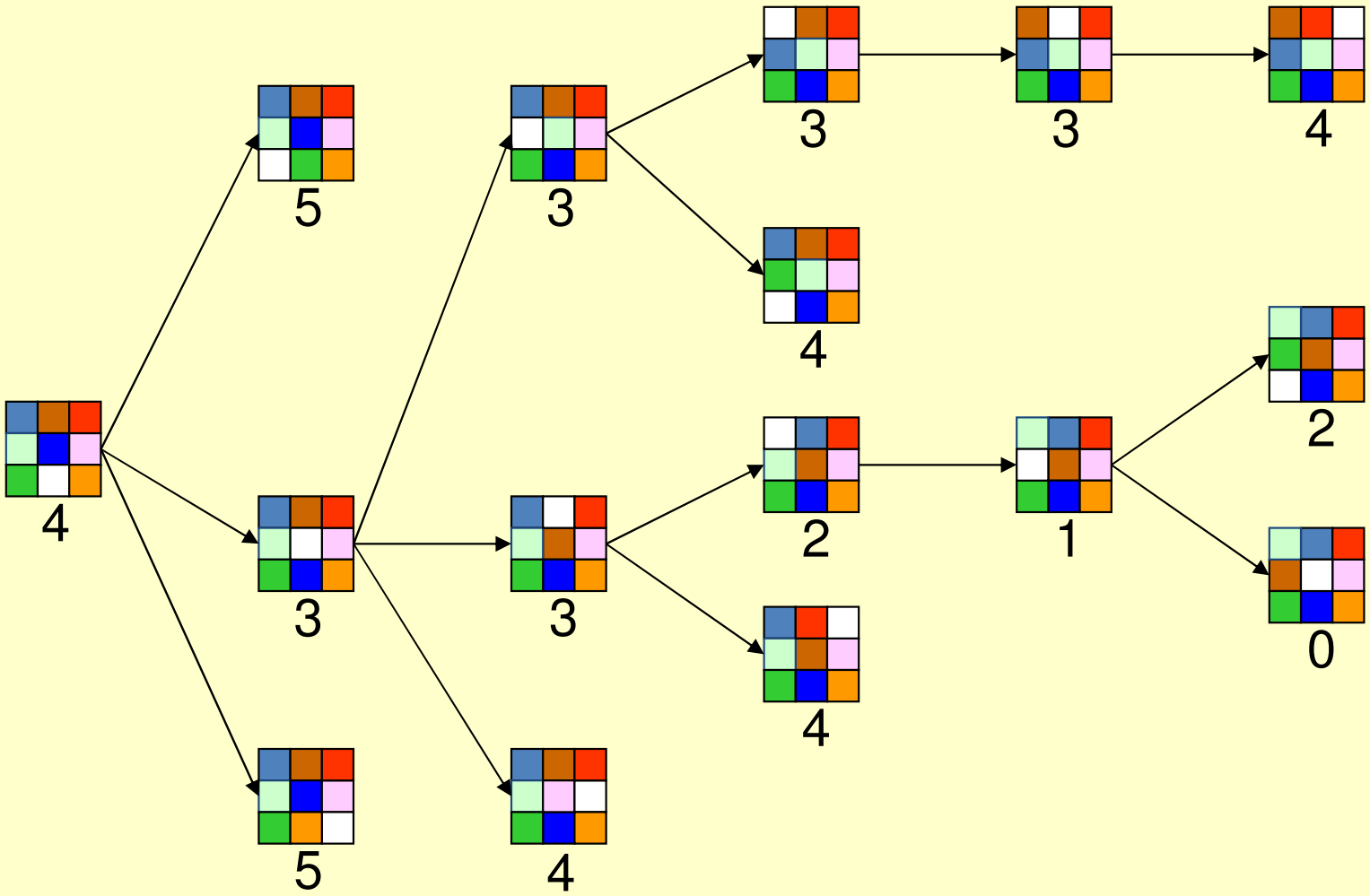


goal

- h1(N) = number of misplaced tiles = 6 → admissible
- h2(N) = sum of distances of each tile to goal = 13 → admissible
- h3(N) = (sum of distances of each tile to goal)
  + (sum of score functions for each tile) = 49 → not admissible

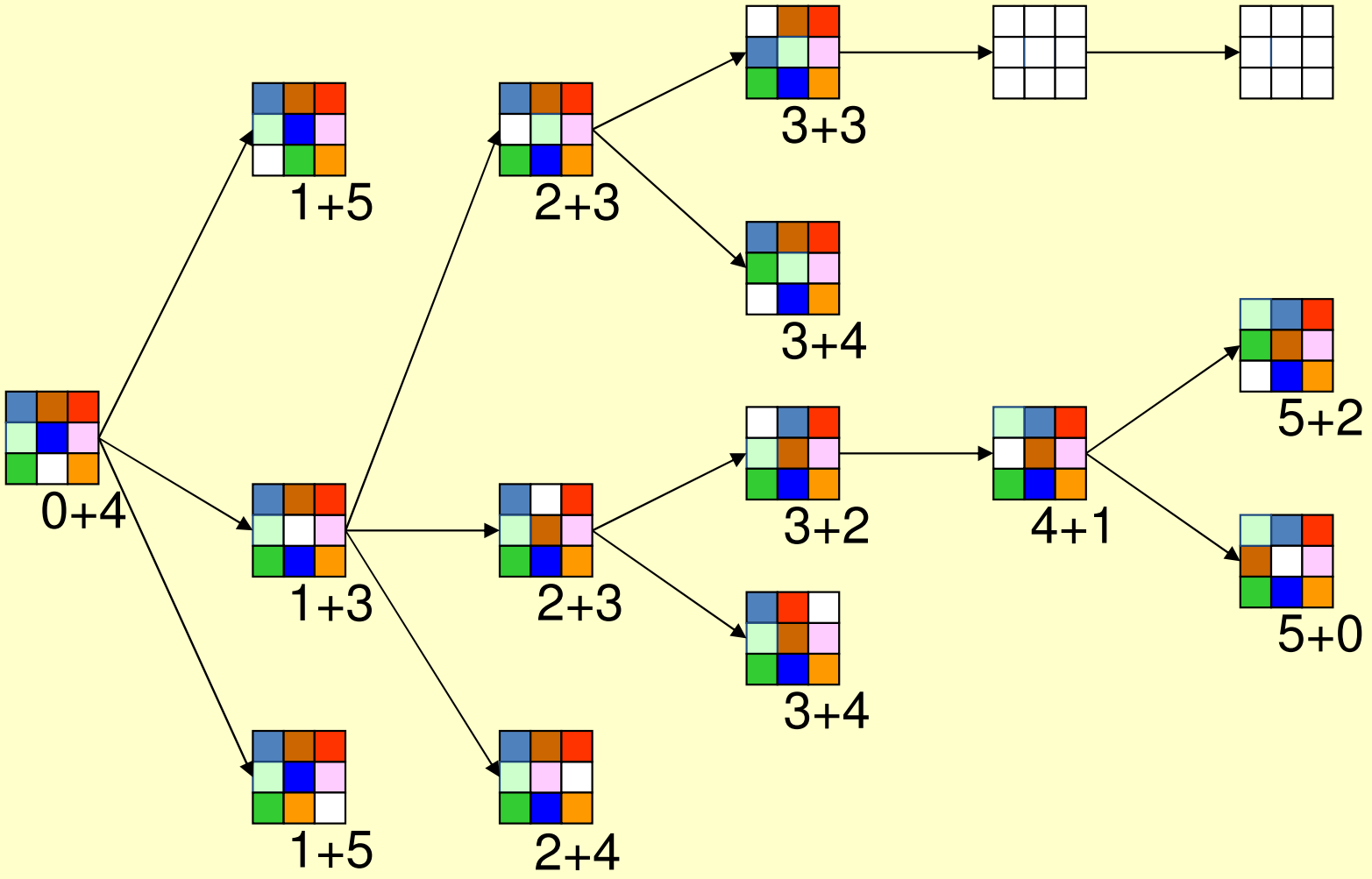
8-Puzzle

f(N) = h(N) = number of misplaced tiles



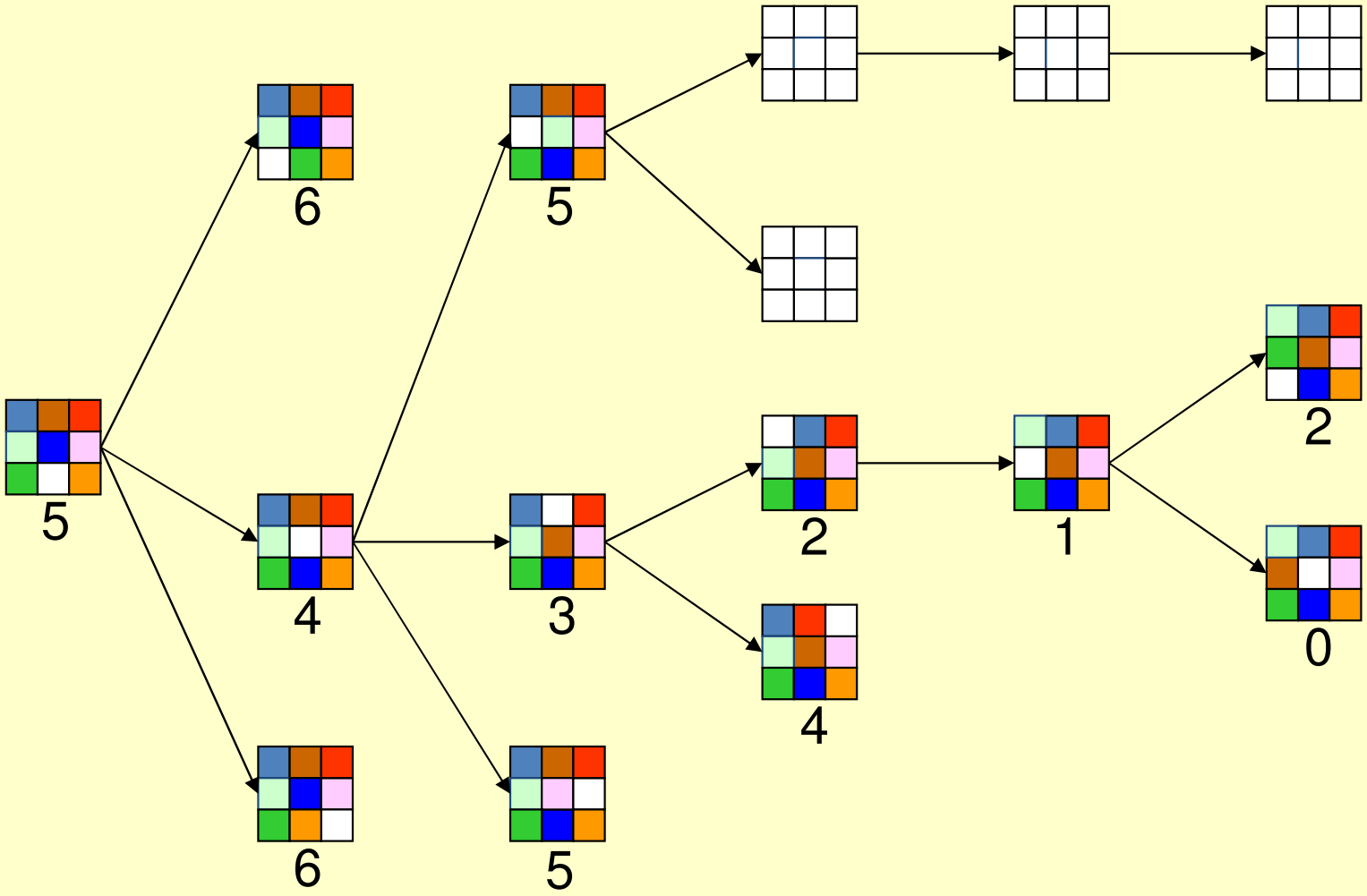
8-Puzzle

f(N) = g(N) + h(N)
with h(N) = number of misplaced tiles



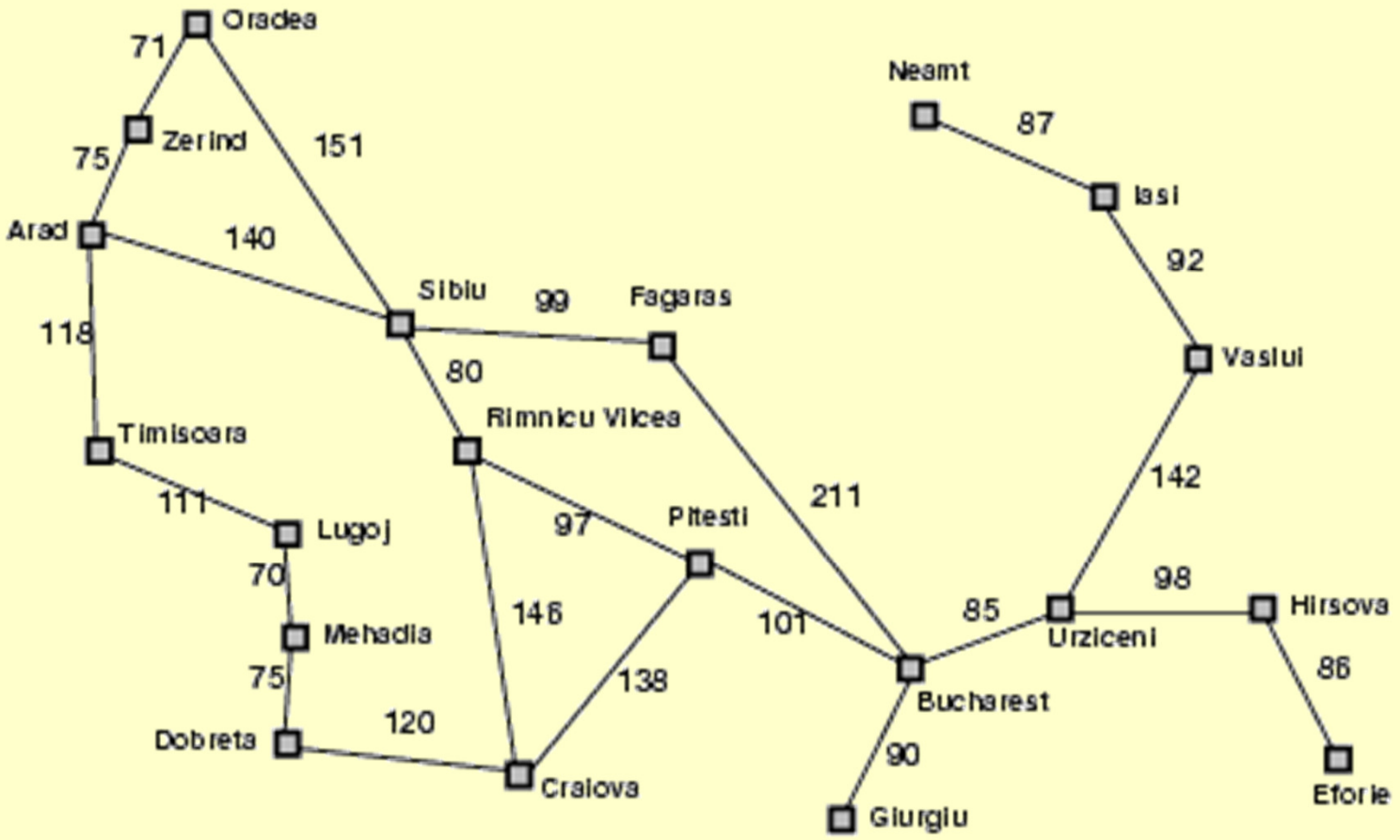
8-Puzzle

f(N) = h(N) = Σ distances of tiles to goal



Map Navigation

Romania with step costs in km



Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374