Informed Search

Sukmawati NE
Departemen Informatika
UNDIP

Informed Search

Non Iteratif, misalnya :

1. **Best-first search**
2. **Greedy best-first search**
3. **A* search**

Iteratif, Misalnya

1. Hill-climbing search
2. Simulated annealing
3. **Genetic algorithms (Dipelajari di akhir kuliah atau MK Evolutionary Algorithm)**

Informed Search

Non Iteratif

1. **Best-first search**
2. **Greedy best-first search**
3. **A* search**

Dipelajari hari ini + Fungsi Heuristic

Iteratif

1. Hill-climbing search
2. Simulated annealing

Dipelajari minggu depan

3. **Genetic algorithms (Dipelajari di akhir kuliah atau MK Evolutionary Algorithm)**

Best-First Search

- Prinsip best-first search : Lakukan node expansion terhadap node di fringe yang nilai f(n)-nya paling kecil.
- Ide dasar : f(n) adalah sebuah **evaluation function** → fungsi yang menyatakan **perkiraan** seberapa “bagus” sebuah node.
- Kenapa perkiraan?

Best-First Search

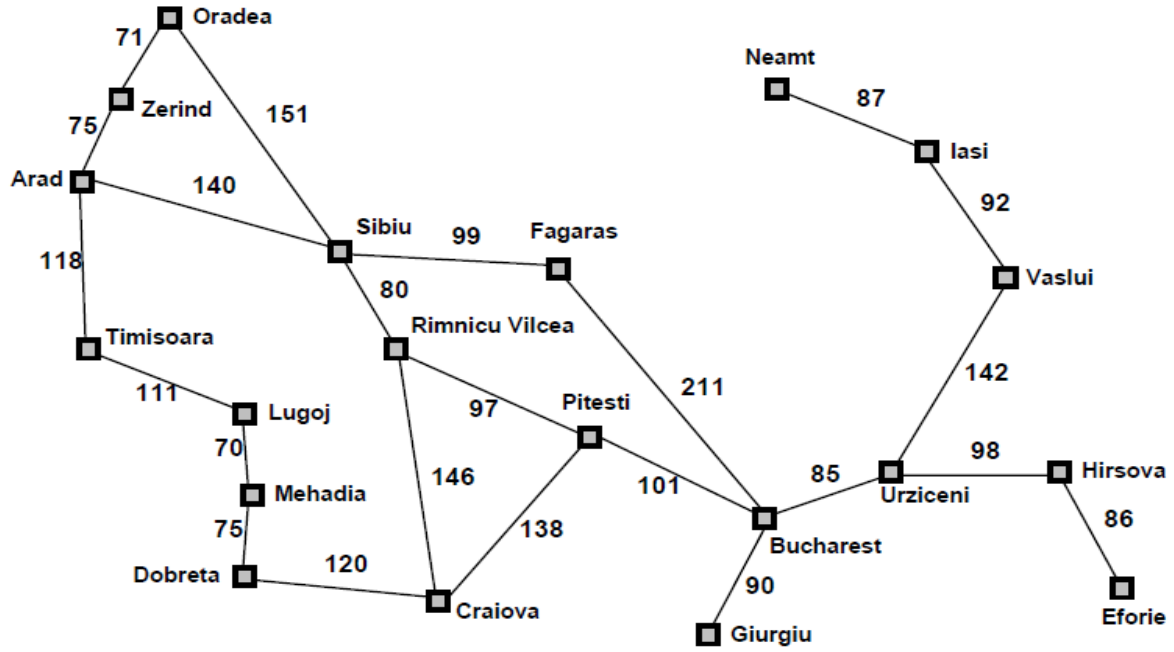
- Prinsip best-first search : Lakukan node expansion terhadap node di fringe yang nilai f(n)-nya paling kecil.
- Ide dasar : f(n) adalah sebuah **evaluation function** → fungsi yang menyatakan **perkiraan** seberapa "bagus" sebuah node.
- Kenapa perkiraan? Kalau tidak, bukan search namanya!
- Implementasi: fringe adalah sebuah priority queue di mana node disortir berdasarkan f(n).
- Contoh :
  - Uniform-cost search
  - Greedy (best-first) search
  - A* search

Heuristic Function

- Kunci keberhasilan best-first search terletak di **heuristic function**.
- Heuristic adalah :
  - **rule of thumb**
  - “kiat-kiat sukses”, “tips-tips keberhasilan”
  - informasi tambahan bagi si agent (agar lebih sukses) → informed search
- Heuristic function h(n) adalah fungsi yang menyatakan estimasi cost dari n ke goal state.
- Ada banyak kemungkinan heuristic function untuk sebuah masalah.

Contoh Heuristic Function

- Romania dengan cost di setiap langkah dalam km



Straight−line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Sebuah heuristic function untuk agent turis Rumania

$h_{SLD}(n)$ = jarak *straight-line distance* dari n ke Bucharest.

Greedy Best-First Search

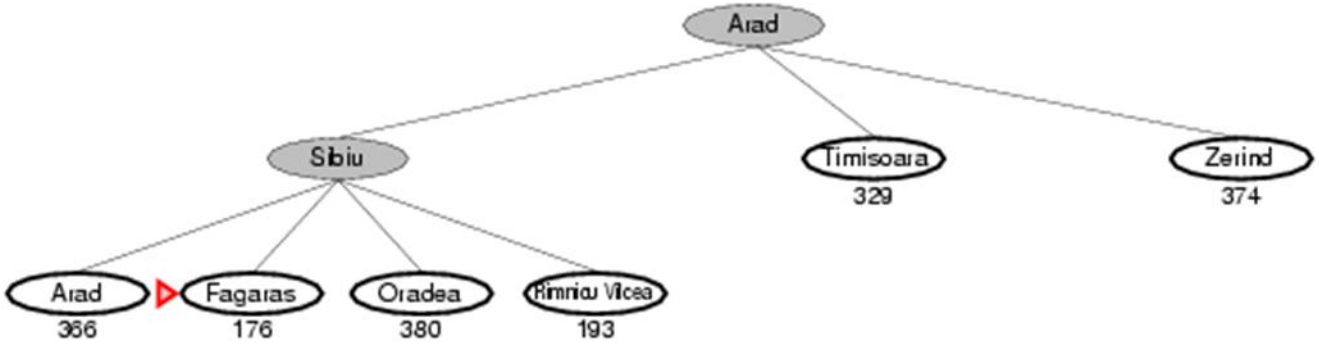
- Prinsip greedy best-first search : Lakukan node expansion terhadap node di fringe yang nilai *h*(n)-nya paling kecil.
- Greedy best-first search selalu memilih node yang **kelihatannya** paling dekat ke goal.

Greedy Best-First Search

- Prinsip greedy best-first search : Lakukan node expansion terhadap node di fringe yang nilai *h*(n)-nya paling kecil.
- Greedy best-first search selalu memilih node yang **kelihatannya** paling dekat ke goal.

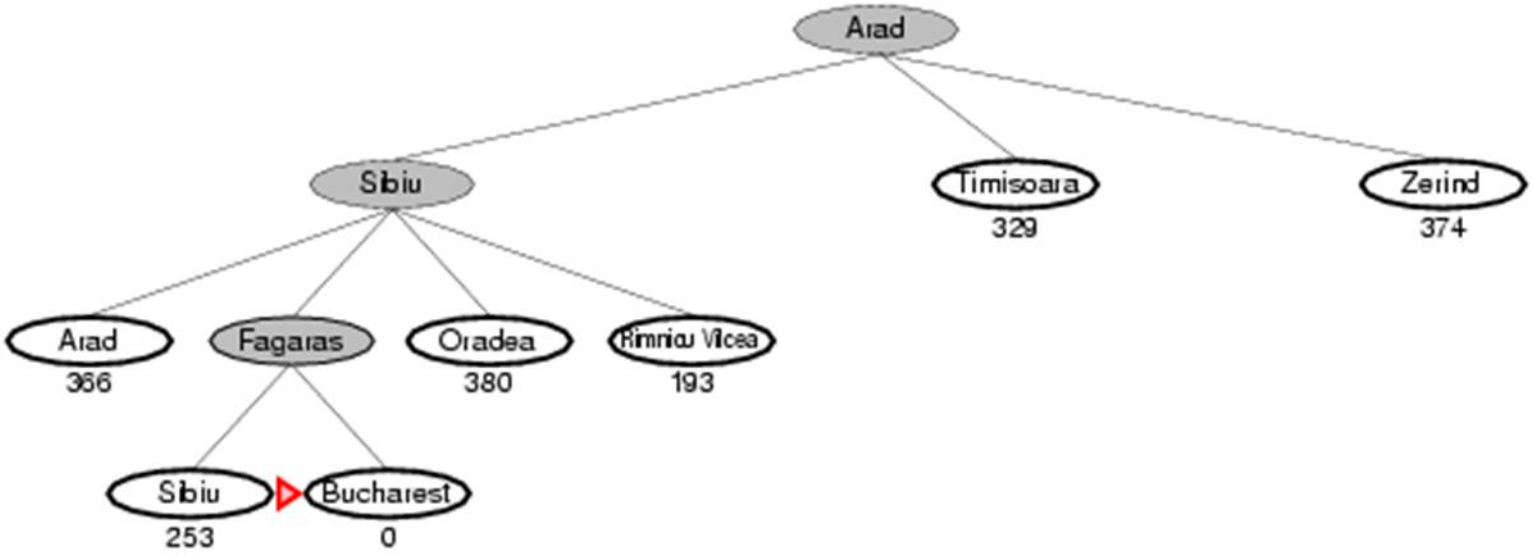
Greedy Best-First Search

- Prinsip greedy best-first search : Lakukan node expansion terhadap node di fringe yang nilai *h*(n)-nya paling kecil.
- Greedy best-first search selalu memilih node yang **kelihatannya** paling dekat ke goal.



Greedy Best-First Search

- Prinsip greedy best-first search : Lakukan node expansion terhadap node di fringe yang nilai *h*(n)-nya paling kecil.
- Greedy best-first search selalu memilih node yang **kelihatannya** paling dekat ke goal.



INGAT KEMBALI!!

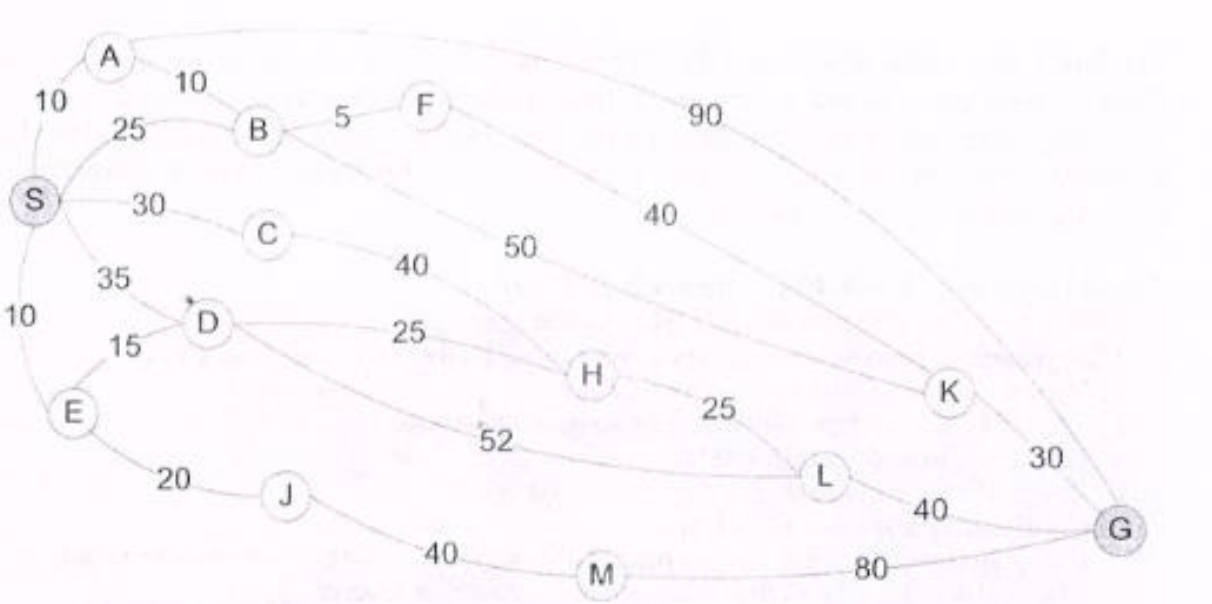
- Search strategy di-evaluasi berdasarkan:
  - completeness: apakah solusi (jika ada) pasti ditemukan?
  - time complexity: jumlah node yang di-generate.
  - space complexity: jumlah maksimum node di dalam memory.
  - optimality: apakah solusi dengan minimum cost pasti ditemukan?
- Time & space complexity diukur berdasarkan
  - b - branching factor dari search tree
  - d - depth (kedalaman) dari solusi optimal
  - m - kedalaman maksimum dari search tree (bisa infinite!)

Greedy Best-First Search

- Properties of greedy best-first search :
  - **Complete**? Ya, jika state space terbatas dan pengulangan state-nya ditangani.
  - **Time complexity**? Secara teoritis, $O(b^m)$, tetapi heuristic function yang baik akan lebih mempercepat.
  - **Space complexity**? $O(b^m)$ → semua node disimpan di memory
  - **Optimal**? Tidak.

Latihan Soal

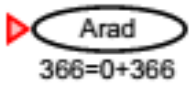
- Dengan Greedy, carilah jarak antara S ke G!
- Apakah hasilnya merupakan jarak terpendek?

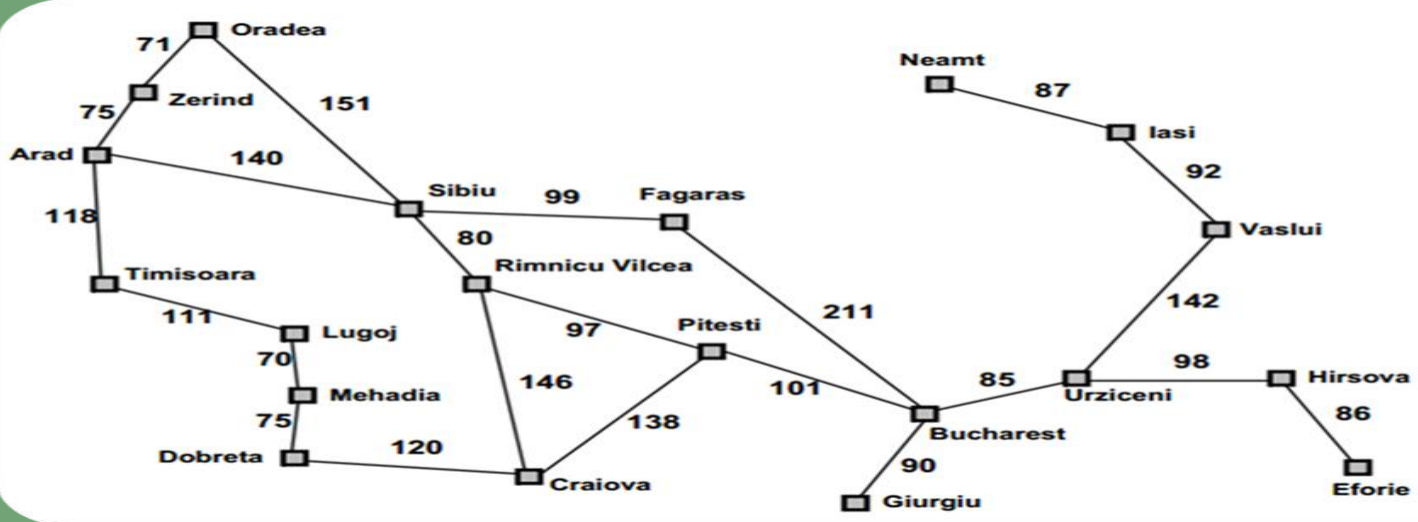


n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h(n)$	80	80	60	70	85	74	70	0	40	100	30	20	70

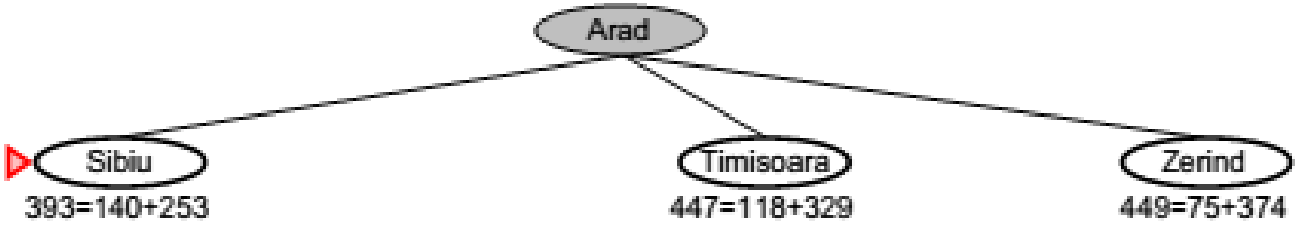
A* Search

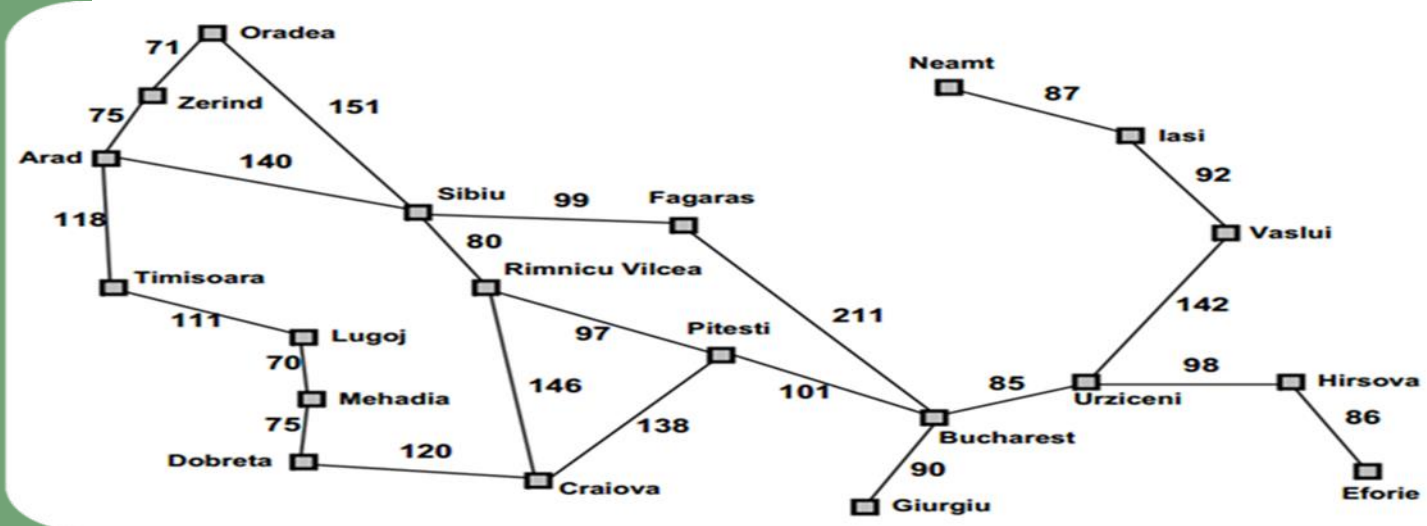
- Prinsip A* search : Hindari node yang berada di path yang “mahal”.
- Evaluation function $f(n) = g(n) + h(n)$:
  - $g(n)$ = Path cost **ke** n
  - $h(n)$ = Estimasi path cost **dari** n ke goal
  - $f(n)$ = Estimasi **total** cost melalui n
- Contoh penelusuran A* searc



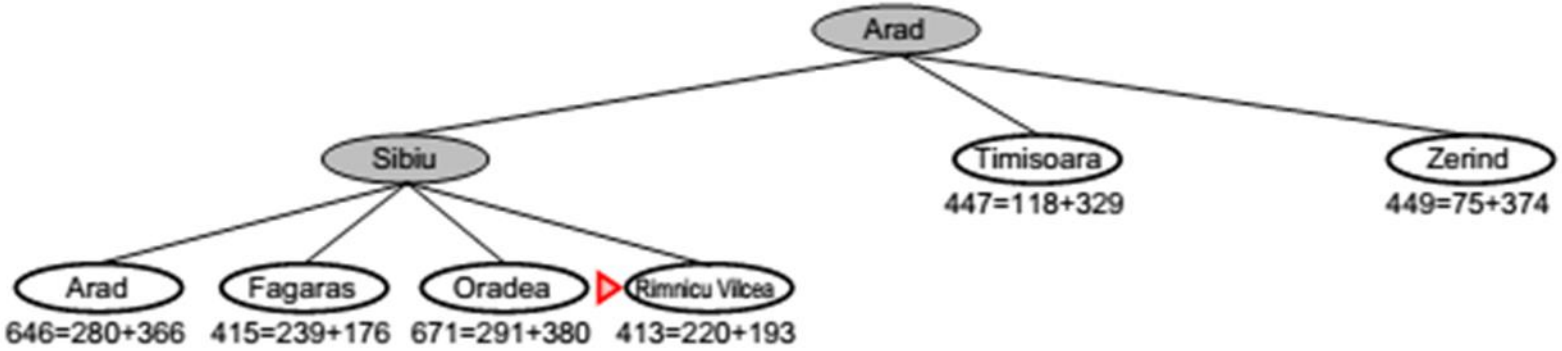


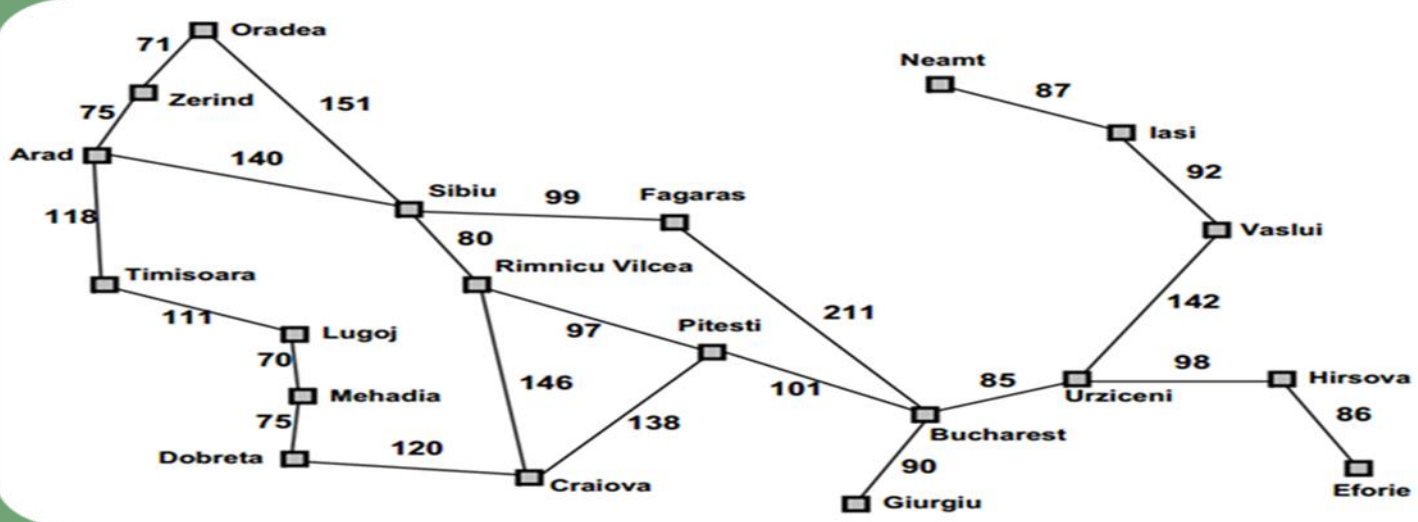
Straight−line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374



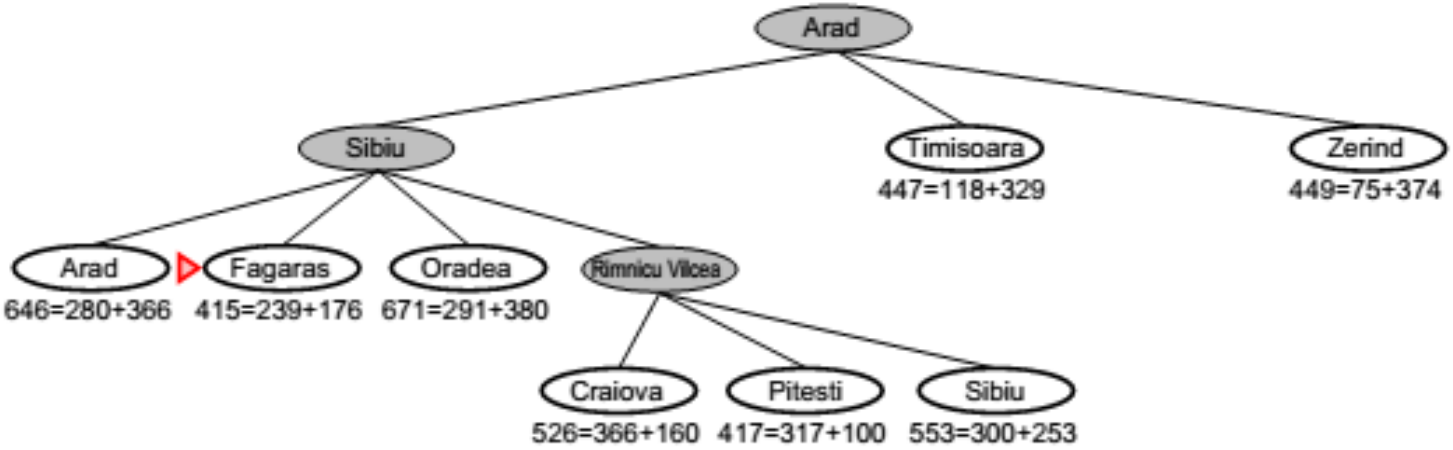


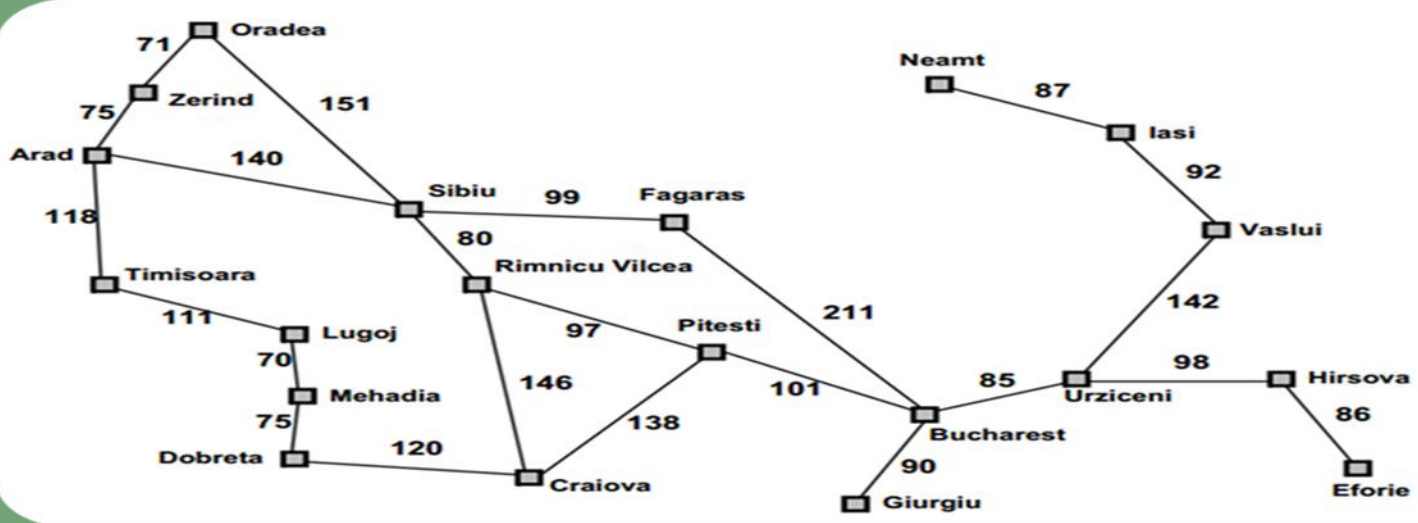
Straight−line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374



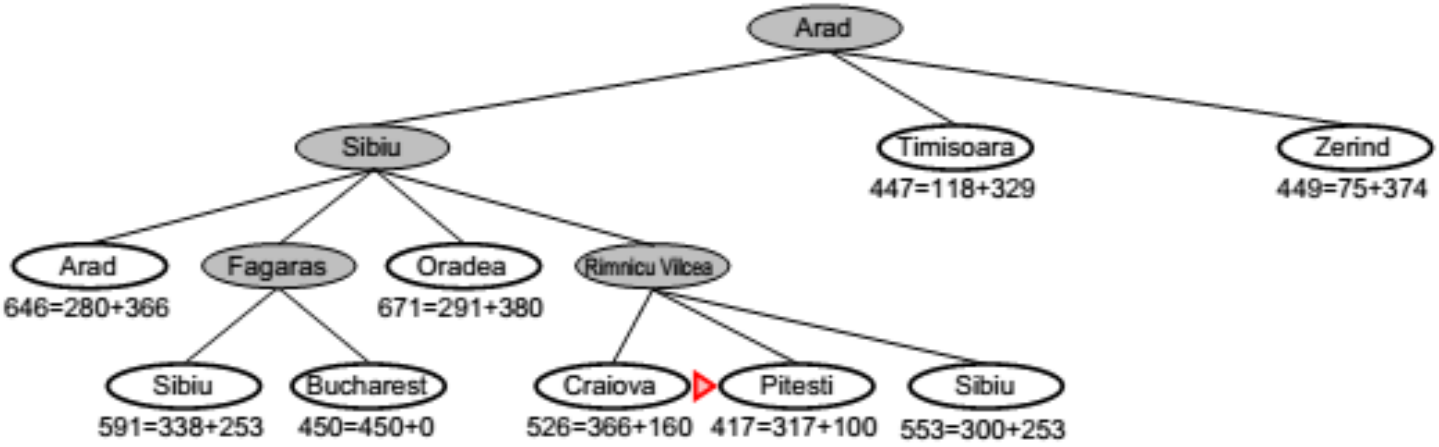


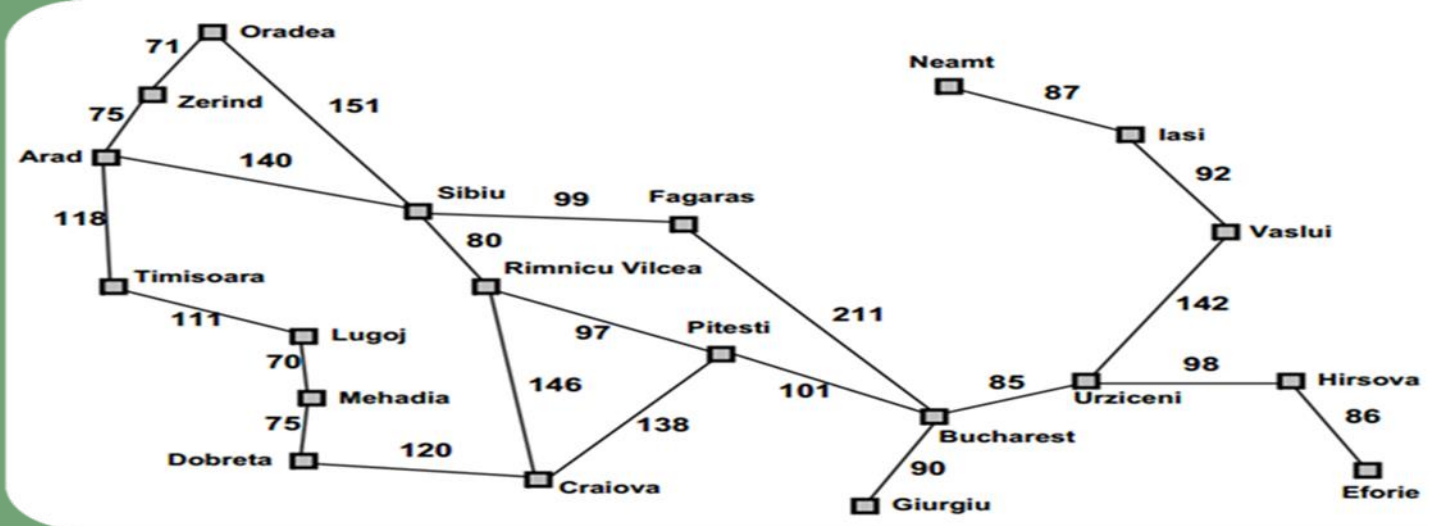
Straight−line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374



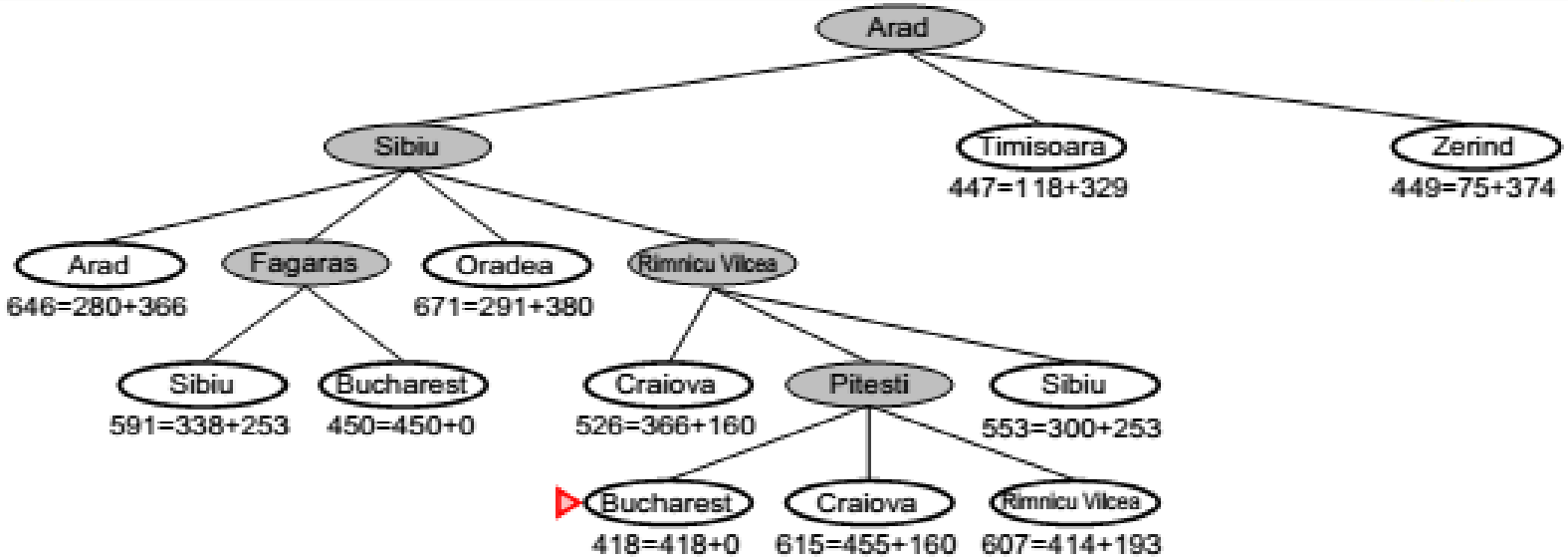


Straight−line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374





Straight−line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

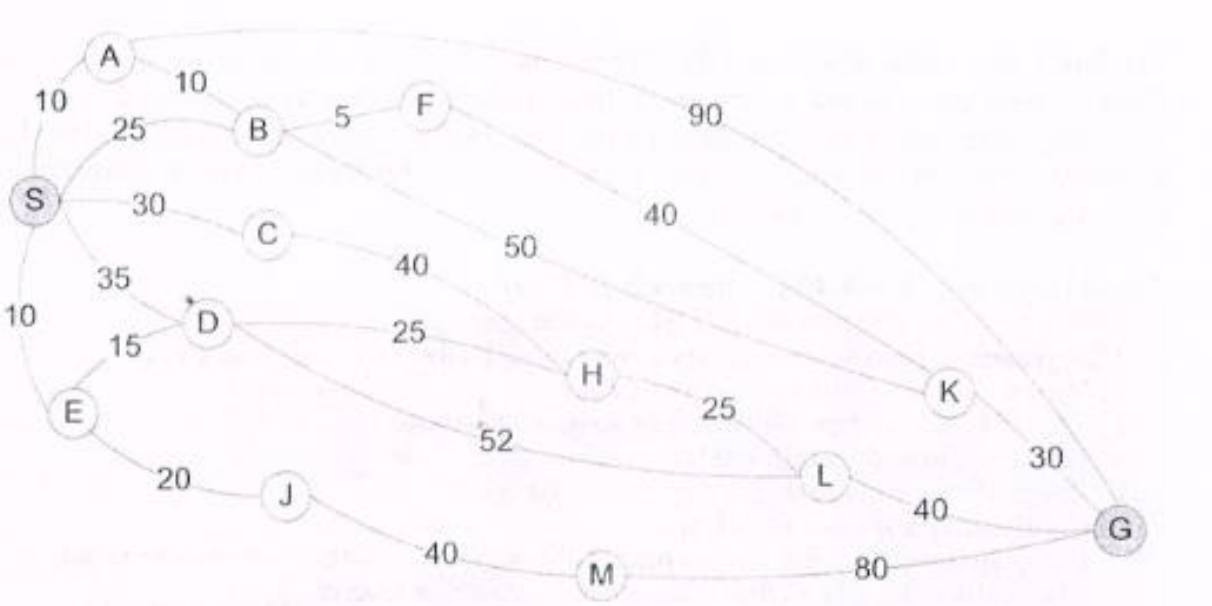


A* Search

- Properties of A* :
  - **Complete**? Ya, kecuali jumlah node di mana f ≤ f(G) tak terbatas.
  - **Time complexity**? Eksponensial (error h x jumlah step solusi).
  - **Space complexity**? $O(b^m)$ → semua node disimpan di memory.
  - **Optimal**? Ya.

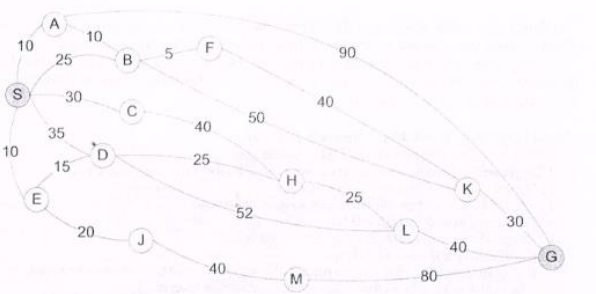
Latihan Soal

- Dengan A*, carilah jarak antara S ke G!
- Apakah hasilnya merupakan jarak terpendek?



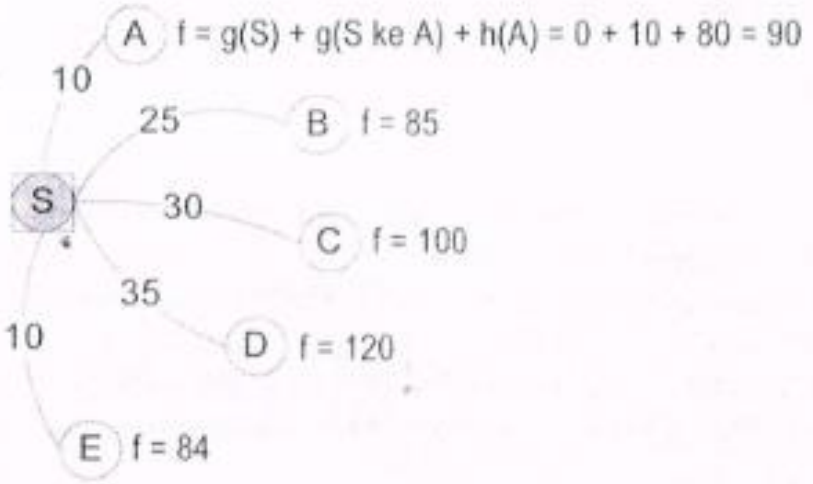
n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h(n)$	80	80	60	70	85	74	70	0	40	100	30	20	70

Jawaban :

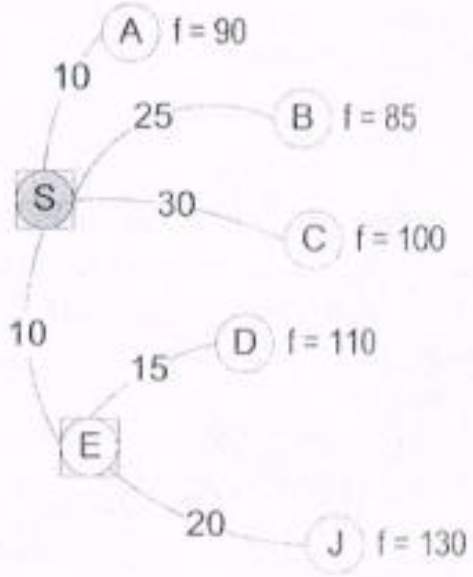


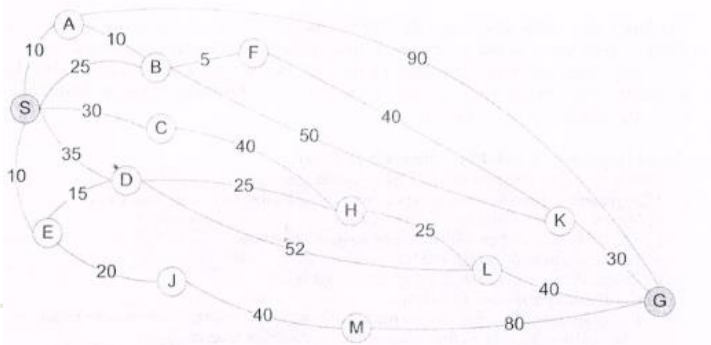
n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h(n)$	80	80	60	70	85	74	70	0	40	100	30	20	70

Langkah 1



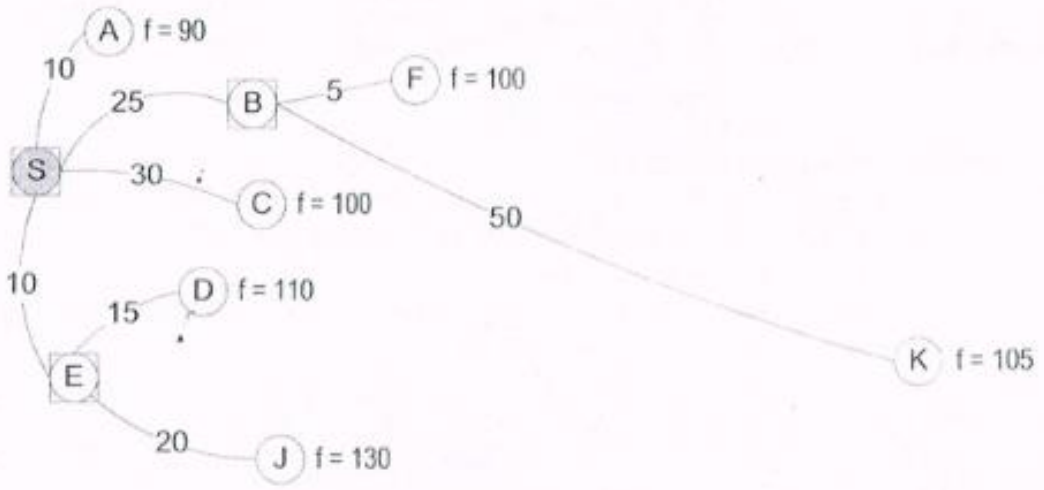
Langkah 2



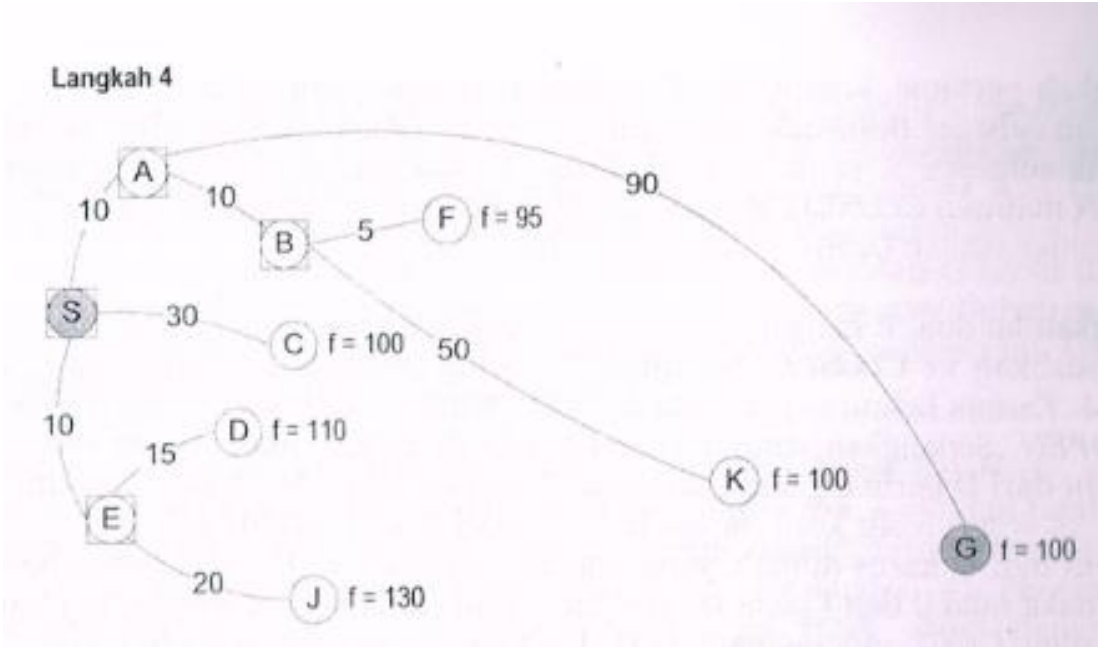


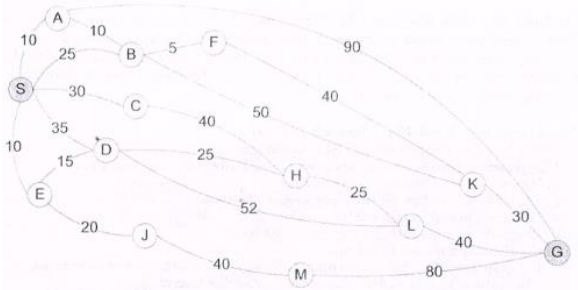
n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h(n)$	80	80	60	70	85	74	70	0	40	100	30	20	70

Langkah 3



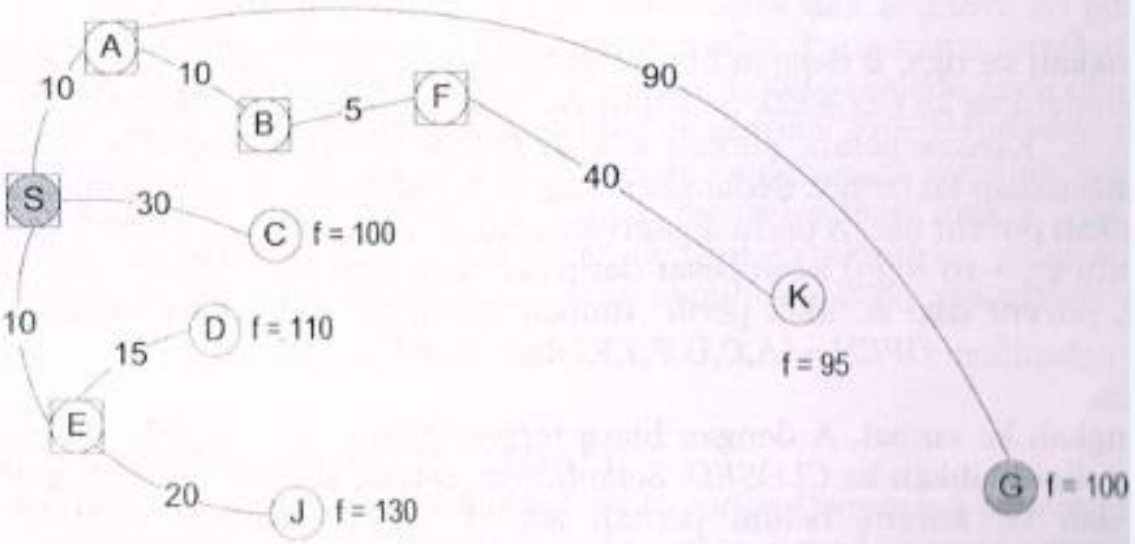
Langkah 4



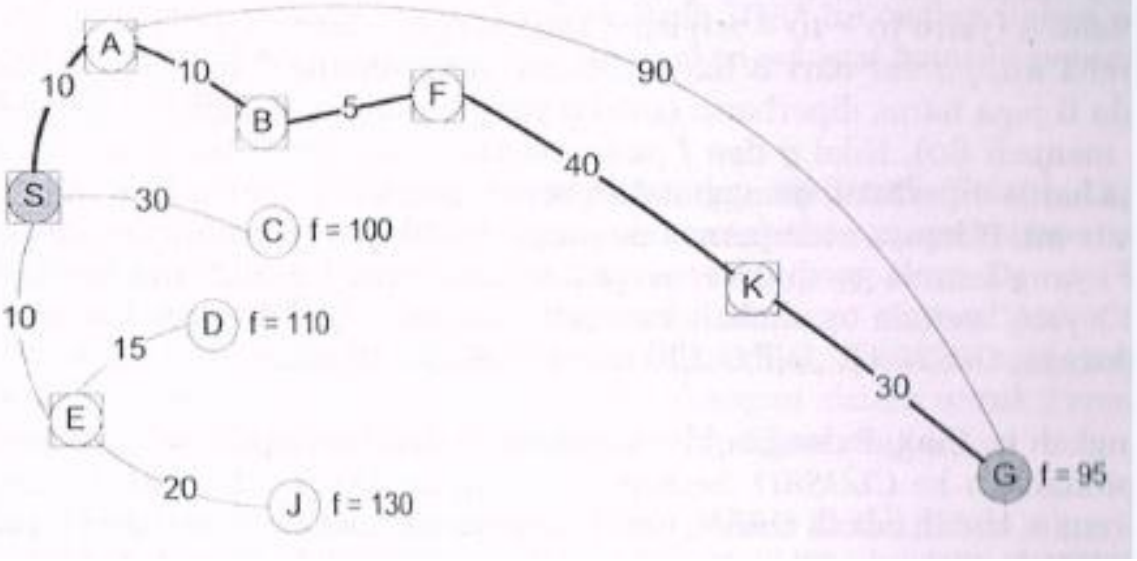


n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h(n)$	80	80	60	70	85	74	70	0	40	100	30	20	70

Langkah 5



Langkah 6



Fungsi Heuristik

- Fungsi Heuristik memainkan peranan yang sangat menentukan

- Suatu fungsi dapat diterima (Admissible) sebagai fungsi Heuristik jika biaya perkiraan yang dihasilkan tidak melebihi dari biaya sebenarnya.

- **Bahasa mudahnya** : nilai sebuah heuristic function tidak pernah **melebihi** cost ke goal yang sebenarnya. Contoh : $h_{SLD}(n)$

Contoh

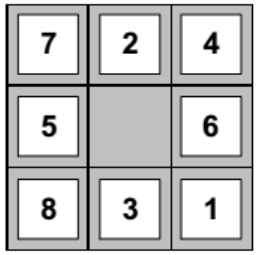
- Masalah Pencarian Rute Terpendek
- Fungsi Heuristik yang digunakan:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

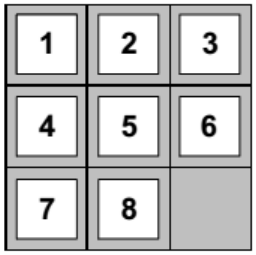
- Fungsi ini akan selalu mendekati, kurang dari atau bahkan sama dengan jarak sebenarnya

Merancang Heuristic

- Contoh Admissible Heuristic
  - h(n) untuk 8-puzzle
    - $h_1(n)$: jumlah angka yang salah posisi.
    - $h_2(n)$: jumlah jarak semua angka dari posisi yang benar (base Manhattan Distance)



Start State



Goal State

(0,0)	(0,1)	(0,2)
(1,0)	(1,1)	(1,2)
(2,0)	(2,1)	(2,2)

Posisi

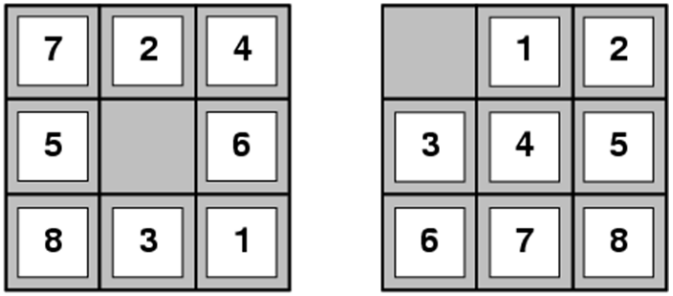
$$D_{man}(x,y) = \sum_{j=1}^{d} |x_j - y_j|$$

    - Diketahui posisi tile 1 di Start State (2,2) dan di Goal State (0,0)
    - Hitung $D_{man}(1) = |2-0|+|2-0| = 2 + 2 = 4$

  - $h_1(s) = 6$
  - $h_2(s) = D_{man}(1) + D_{man}(2)+ D_{man}(3) + D_{man}(4)+ D_{man}(5) + D_{man}(6)+ D_{man}(7) + D_{man}(8)= 4+0+3+3+1+0+2+1=14$

Merancang Heuristic

- Latihan Admissible Heuristic
  - Perhatikan 8-puzzle berikut :



$$D_{man}(x,y) = \sum_{j=1}^{d} \left| x_j - y_j \right|$$

Tentukan $h_1(s)$ dan $h_2(s)$!

Berapakah cost (biaya) sebenarnya untuk menyelesaikan masalah ini??

Merancang Heuristic

- Kedua fungsi diatas dapat digunakan sebagai fungsi heuristik karena total langkah yang diperlukan dari satu state menuju goal state minimal sama dengan jumlah posisi yang salah (h1) atau sama dengan total langkah yang diperlukan masing-masing kotak untuk menuju posisi yang benar di goal state.

- h2 >= h1, manakah yang lebih baik??

Merancang Heuristic

- Membandingkan dua heuristic
  - h_1 dan h_2 sama-sama admissible. Mana yang **lebih baik**? Bandingkan jumlah node yang di-expand:

d	IDS	A*(h_1)	A*(h_2)
12	3,473,941	539	113
24	54,000,000,000	39,135	1,641

  - d : depth, IDS : Iterative Deepending Search
  - Jika $h_2(n) \geq h_1(n)$ untuk semua n (dan keduanya admissible), dikatakan bahwa h_2 men-***dominate*** h_1 dan lebih baik untuk search.
  - Semakin besar nilai h(n), semakin dekat ke h*(n), semakin banyak node yang **tidak di-expand** (di-**prune**), semakin efisien search-nya!