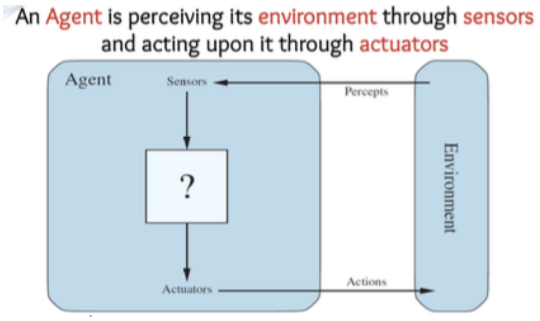
- Definisi AI
    - Kemampuan mesin untuk imitate intelligent human behavior
    - Teori dan pengembangan computer system yang mampu melakukan tugas yang biasanya membutuhkan human intelligence
    - Ilmu dan rekayasa untuk membuat intelligent machines
    - Rational agent yang melakukan hal yang benar

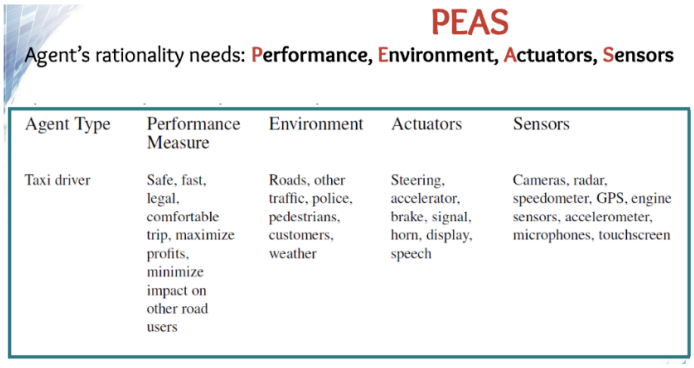
- Empat Jenis AI menurut Russel Norvig
    a. Thinking humanly = mengembangkan teori tentang cara kerja pikiran manusia
    b. Acting humanly = menggunakan Turing Test, questioner decide which is human and which is computer
    c. Thinking rationally = dapat menyelesaikan masalah apa pun yang dapat dipecahkan yang dijelaskan dalam pengertian logis
    d. Acting rationally = achieve the best outcome or the best expected outcome

- Konsep tentang Agent dan Rational Agent
    a. Agent
        - Mengamati environment melalui sensors and bertindak atas hal tersebut melalui actuators
        - Percept = perceptual inputs pada saat tertentu
        - Percept sequence = complete history of percepts
        - Agent's behavior = fungsi yang memetakan percept sequence to an action
        - Agent program = implementation of agent function
        - Contoh = human, robot, vacuum cleaner, cellphone, etc



    b. Rational Agent
        - Should select an action
        - Maximize its performance measure
        - Being rational = maximizing your expected utility

- PEAS dan Task Environment



Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits, minimize impact on other road users	Roads, other traffic, police, pedestrians, customers, weather	Steering, accelerator, brake, signal, horn, display, speech	Cameras, radar, speedometer, GPS, engine sensors, accelerometer, microphones, touchscreen

a. Observability
   - Fully observable = sensor give access to the complete state of environment
   - Partially = give access to the partial state environment
   - Unobservable = give no access to the state

b. Agents
   - Only 1 agent
   - 2 or more agents. Competitive (maximize its performance and minimize others), Cooperative (maximize its performance and others)

c. Deterministic, Stochastic, Strategic
   - Deterministic = next state of environment is **completely** determined by the current state
   - Stochastic = next state of the environment **cannot be** determined by the current state
   - Strategic = deterministic, except for the action of other agents

d. Episodicity
   - Episodic = next episode tidak tergantung pada action taken on prev episode
   - Sequential = current decision bisa mempengaruhi semua future decisions

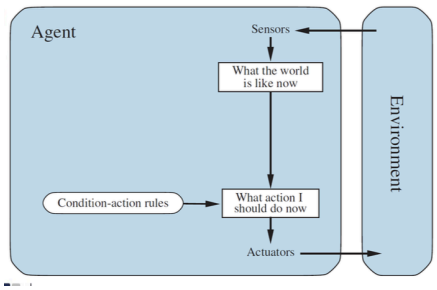
e. Static vs Dynamic
   - Static = environment tidak berubah
   - Dynamic = environment dapat berubah saat agent berpikir
   - Semi-Dynamic = environment doesn't change with time but agent's performance does

f. Continuity
   - Discrete = finite number of different states
   - Continuous = actions have range of continuous values

g. Known vs Unknown
   - Refers pada status pengetahuan agen tentang law of physics of environment
   - Known = outcomes for all action are given
   - Unknown = outcomes aren't given, agent have to learn how it works

Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

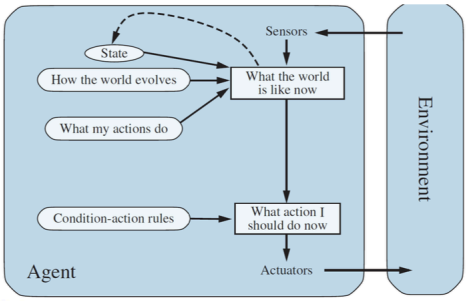
- Lima Jenis Struktur Agents
  a. Simple Reflex Agents
     - Condition-Action rule
     - Simple tetapi limited intelligence
     - Unobservability = menyebabkan infinite loop, tetapi can be solved with randomize



function SIMPLE-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *rules*, a set of condition–action rules

  state ← INTERPRET-INPUT(*percept*)
  rule ← RULE-MATCH(*state*, *rules*)
  action ← *rule*.ACTION
  return *action*

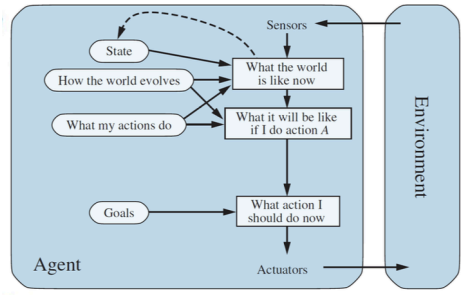
b. Model-based Reflex Agents
   - Partial observability = maintain internal state untuk memodelkan unobserved aspect of state saat ini
   - Transition model = next state depends on the current state and action
   - Sensor model = how the state of the world tercermin dalam agent's percept



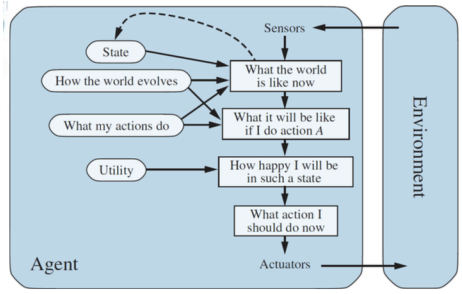
function MODEL-BASED-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *state*, the agent's current conception of the world state
    transition_model, a description of how the next state depends on the current state and action
    sensor_model, a description of how the current world state is reflected in the agent's percepts
    rules, a set of condition–action rules
    action, the most recent action, initially none

state ← UPDATE-STATE(*state*, *action*, *percept*, *transition_model*, *sensor_model*)
rule ← RULE-MATCH(*state*, *rules*)
action ← *rule*.ACTION
return *action*

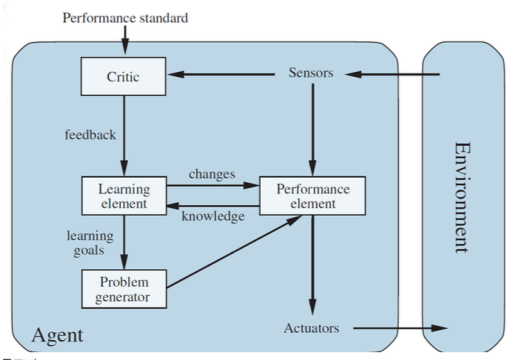
c. Goal-based Agents
   - Current state tidak cukup untuk decide action yang harus diambil
   - Goal = menjelaskan situasi yang diinginkan
   - Agent dapat combine goal dengan previous model to choose action yang akan mencapai goal
   - Searching and planning = find action sequence that achieve agent's goal



d. Utility-based Agents
   - Goals saja tidak cukup to generate high-quality behavior
   - Utility function = rasional menurut external performance measure
   - Choose action yang memaksimalkan utility yang diharapkan dari action outcomes

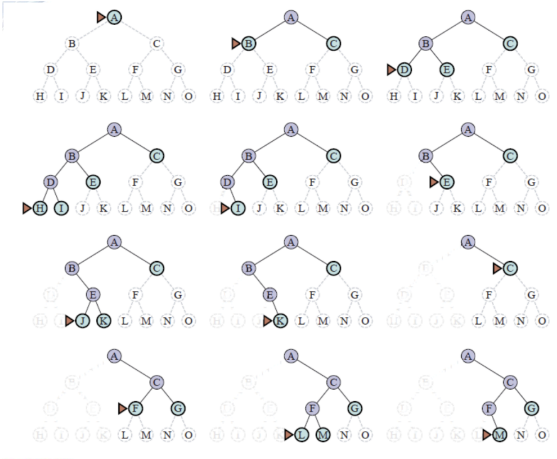


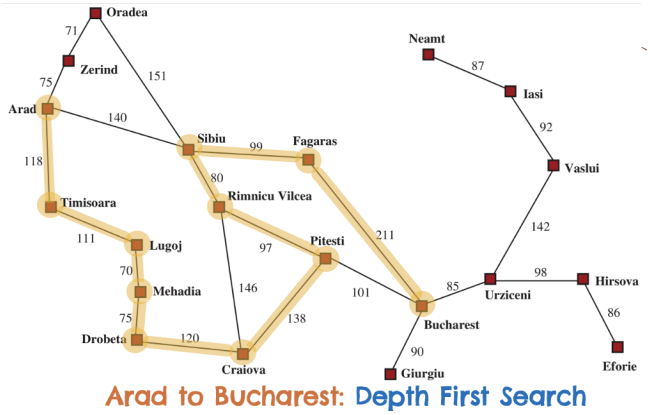
e. Learning Agents
   - Memungkinkan agen to operate in unknown environment dan menjadi lebih kompeten daripada initial knowledge
   - Four conceptual elements
      - Learning element = making improvement
      - Performance element = selecting external actions
      - Critic gives feedback on how agent is doing
      - Problem generator = suggesting actions yang akan mengarah pada exploratory experiments



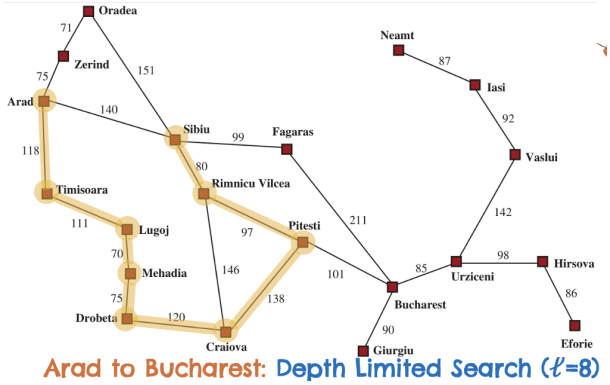
- Konsep tentang Uninformed dan Informed Search
   - Uninformed = no information about the goals
   - Informed = menggunakan domain-specific hints about the goals

- Uninformed Search
   a. DFS (Depth-First Search)
      - Expand terlebih dahulu unexpanded node terdalam

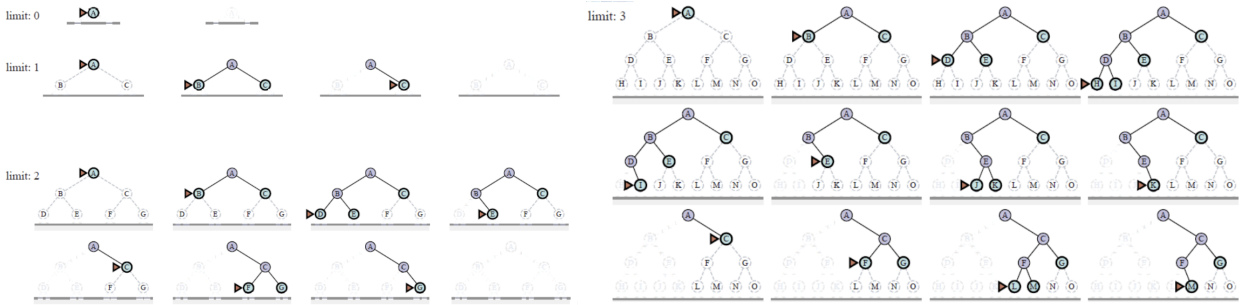




b. DLS (Depth-Limited Search)
   - DFS dengan limited depth (avoid infinite path)



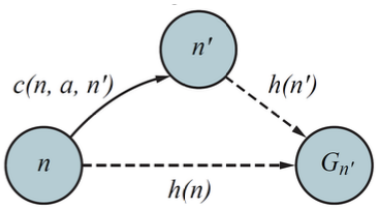
c. ID (Iterative Deepening)
   - DLS dengan increasing value



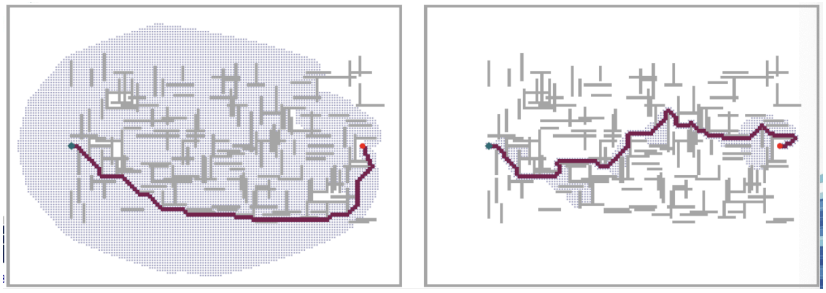
- Informed Search
  a. Greedy BFS
     - Expands node dengan lowest value of h(n)
     - f(n) = h(n)
  b. A* Search
     - BFS dengan admissible & consistent heuristic
     - f(n) = g(n) + h(n)

- Heuristic Function
  - Consistent pasti admissible, tidak sebaliknya
  a. Admissible
     - Never overestimates the cost to reach a goal
     - Cost lebih besar dari h()

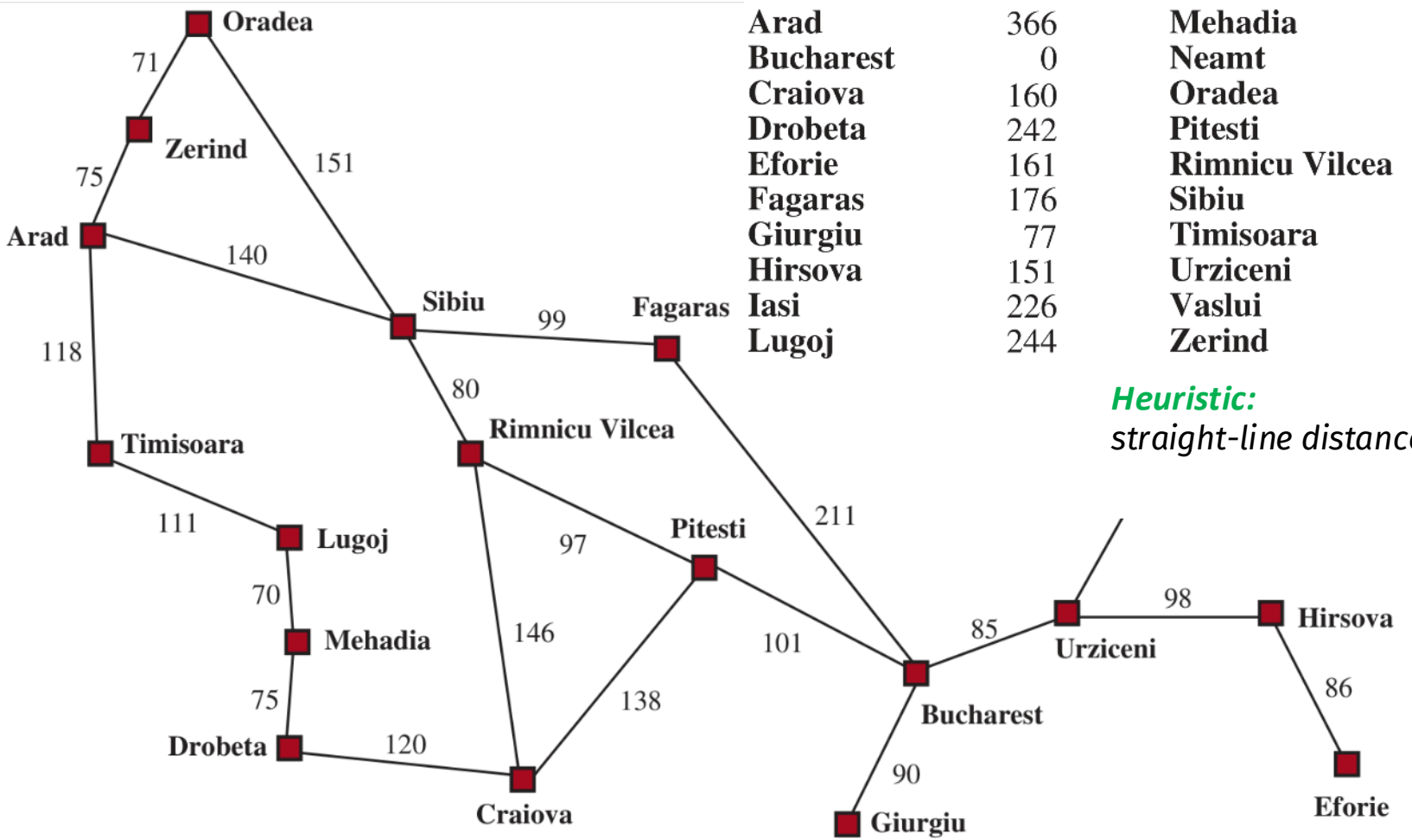
b. Consistent
   - Satisfy the triangle inequality
   - h(n) <= c(n, a, n') + h(n')



- Weighted A* Search
  - Consistent heuristic is optimally efficient
  - Weight the heuristic value more heavily
  - f(n) = g(n) + W x h(n) (1 < W < infinite)



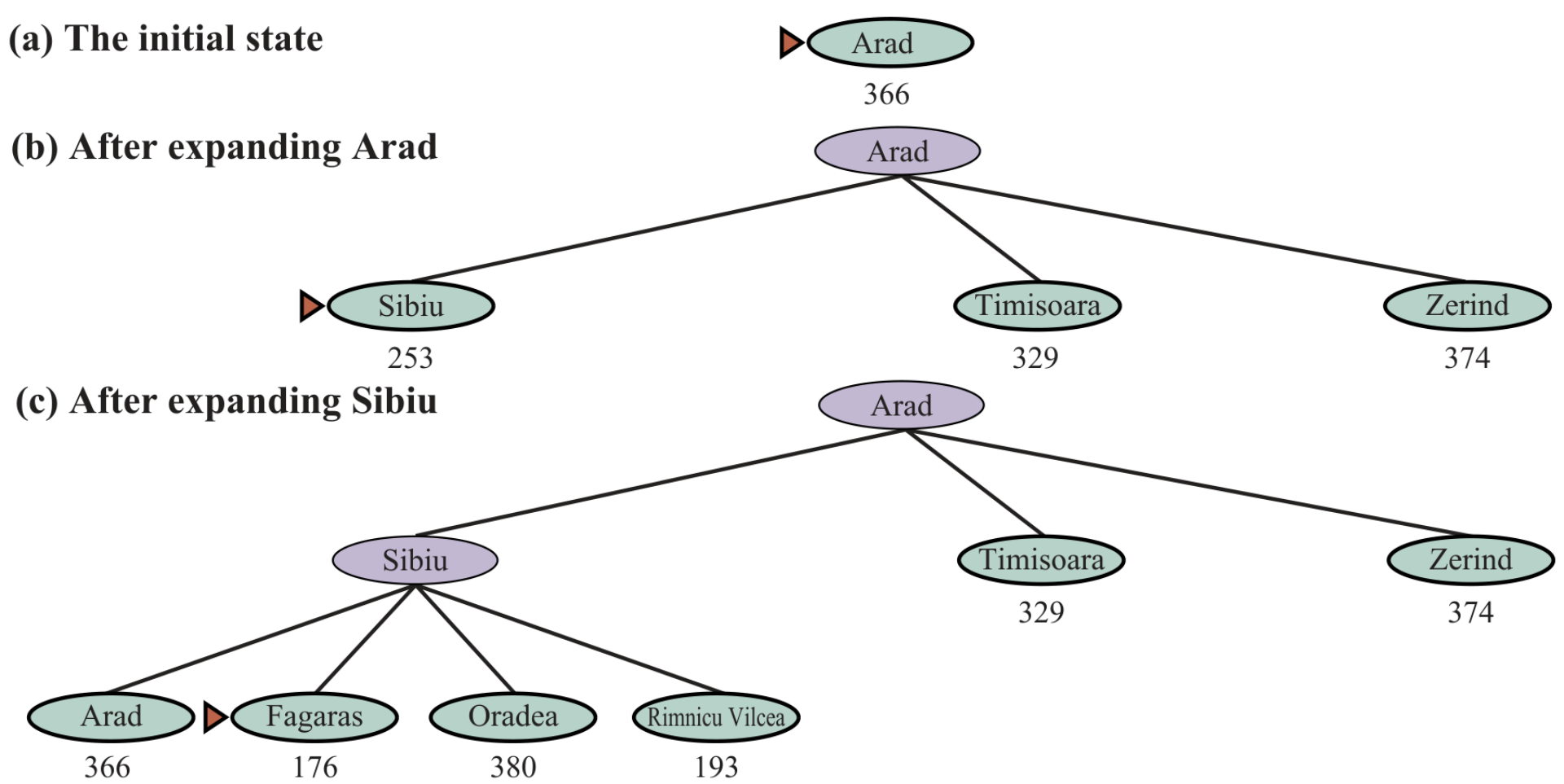
- Beam Search
  a. Simpan k node dengan best f-scores, abaikan expanded node
     - Tidak lengkap dan suboptimal
     - Cepat, karena memperluas lebih sedikit node
  b. Versi lain: tidak membatasi k, sebaliknya simpan setiap node yang skor-f-nya berada dalam skor-f terbaik



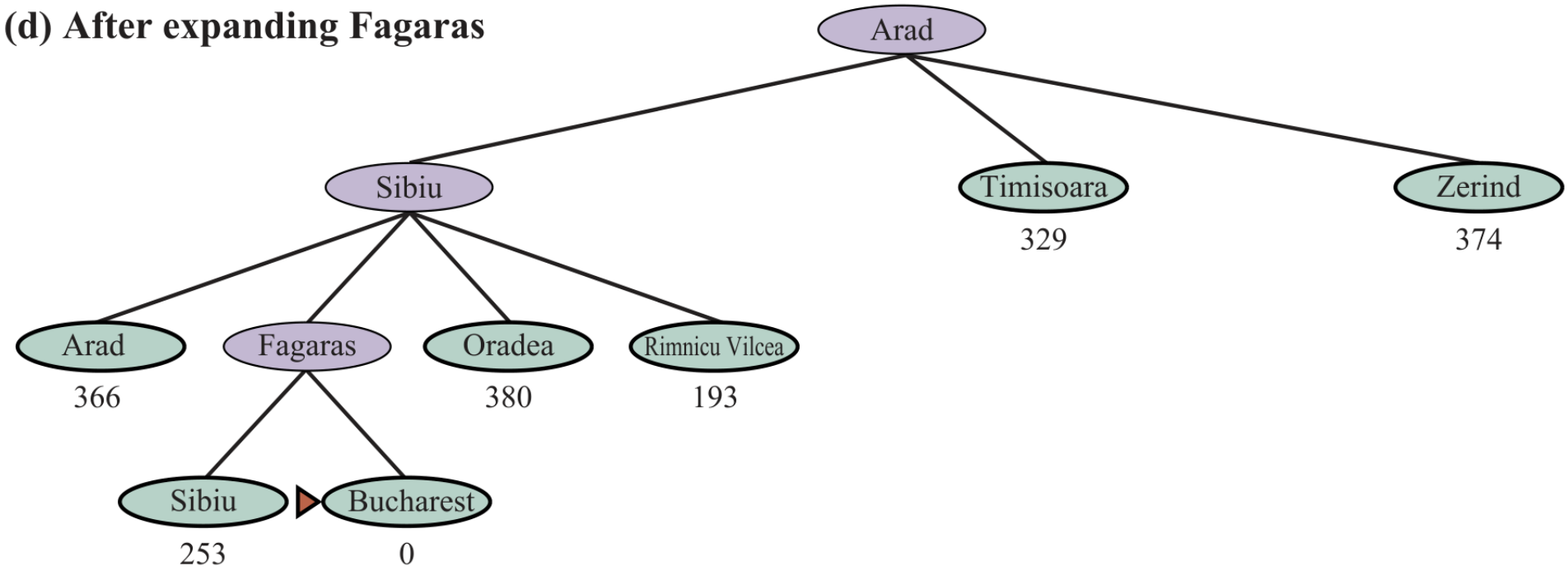
Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

Heuristic:
straight-line distance

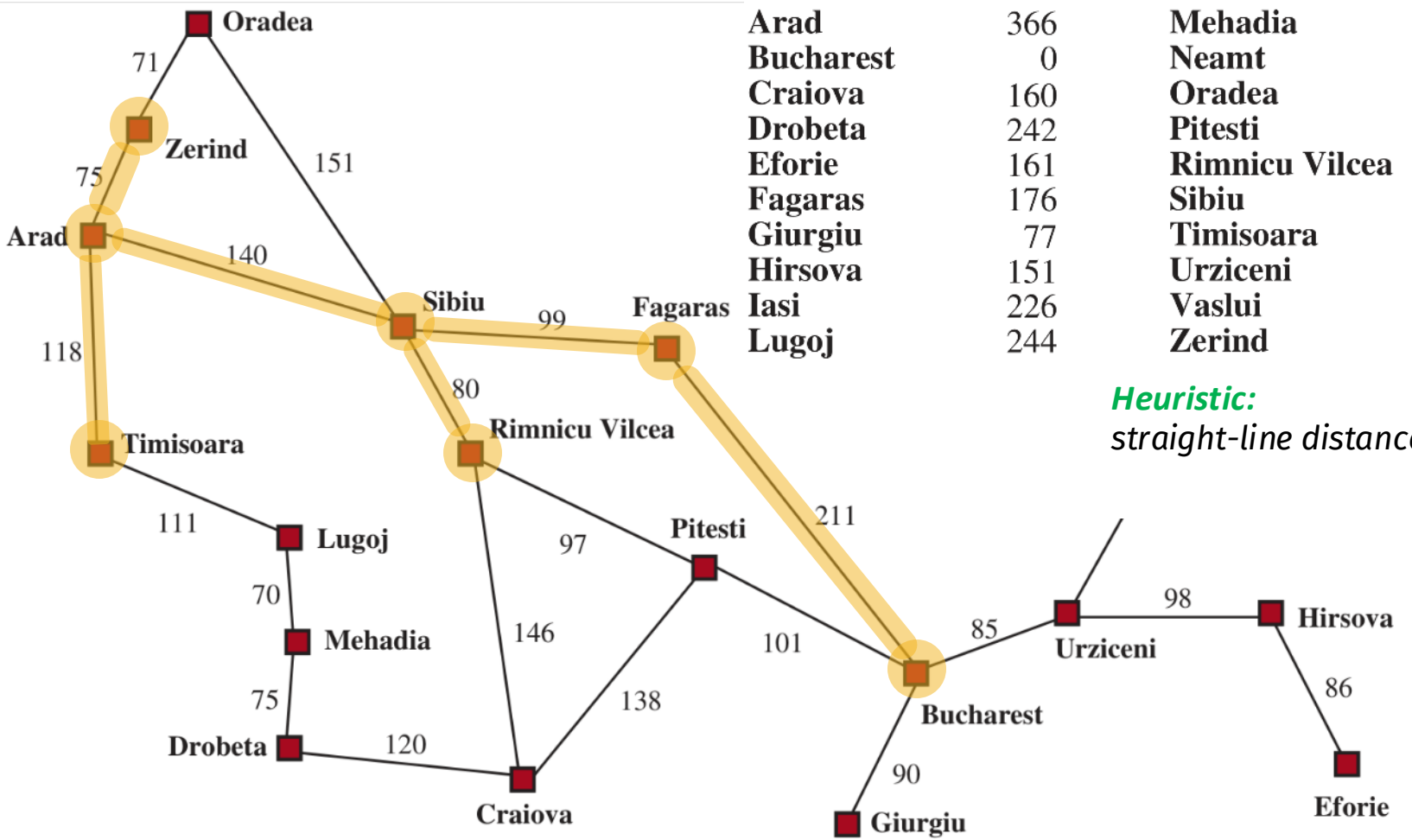
Arad to Bucharest: Greddy Best First Search



Arad to Bucharest: Greddy Best First Search



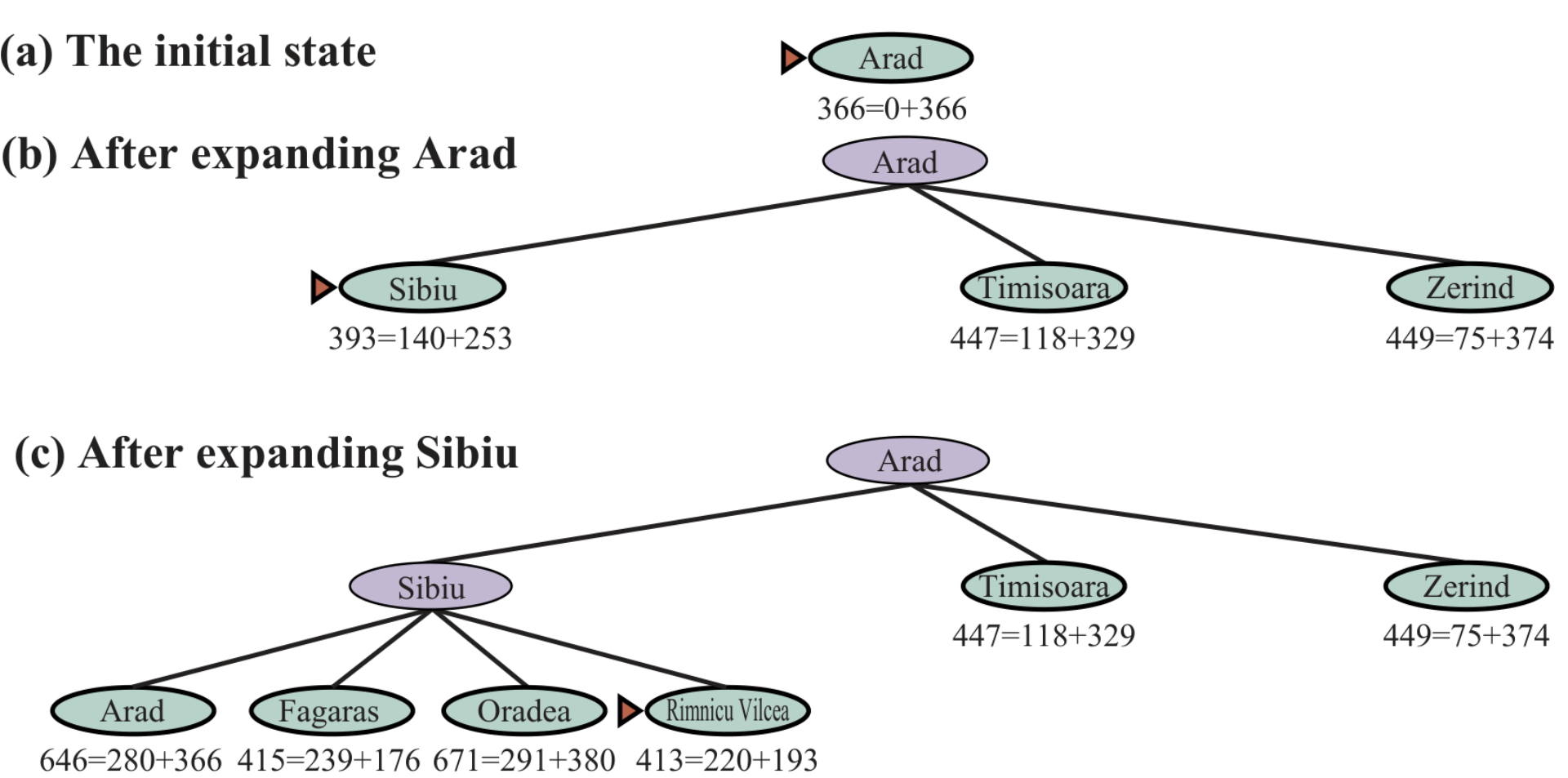
Arad to Bucharest: Greddy Best First Search



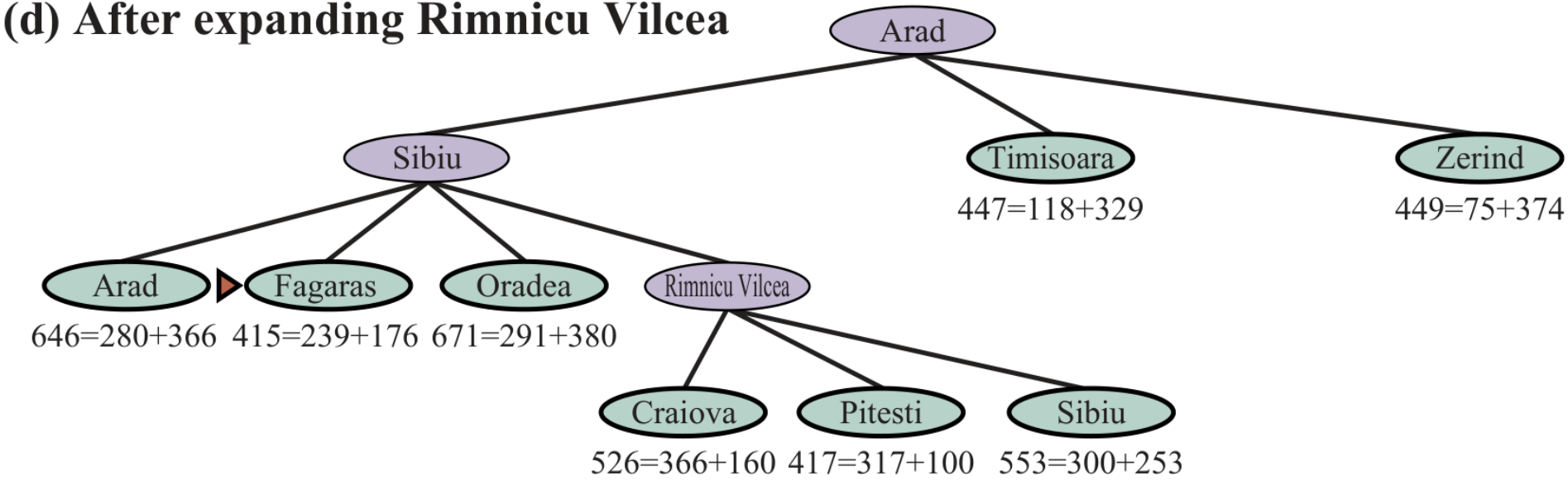
Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

Heuristic:
straight-line distance

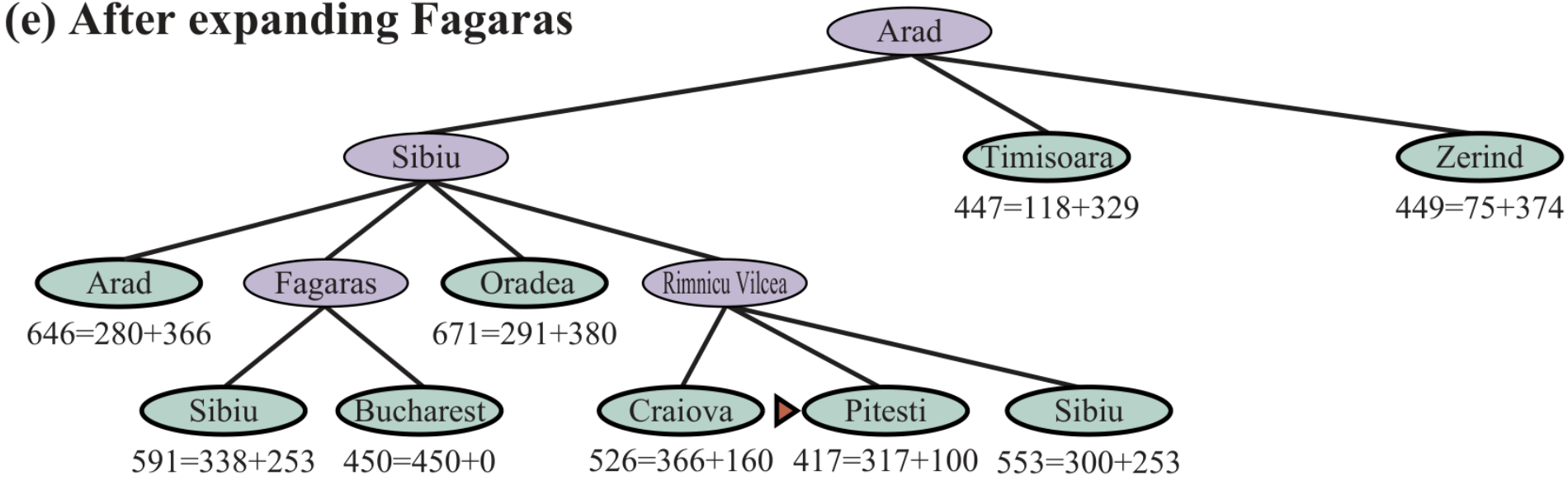
Arad to Bucharest: Greddy Best First Search



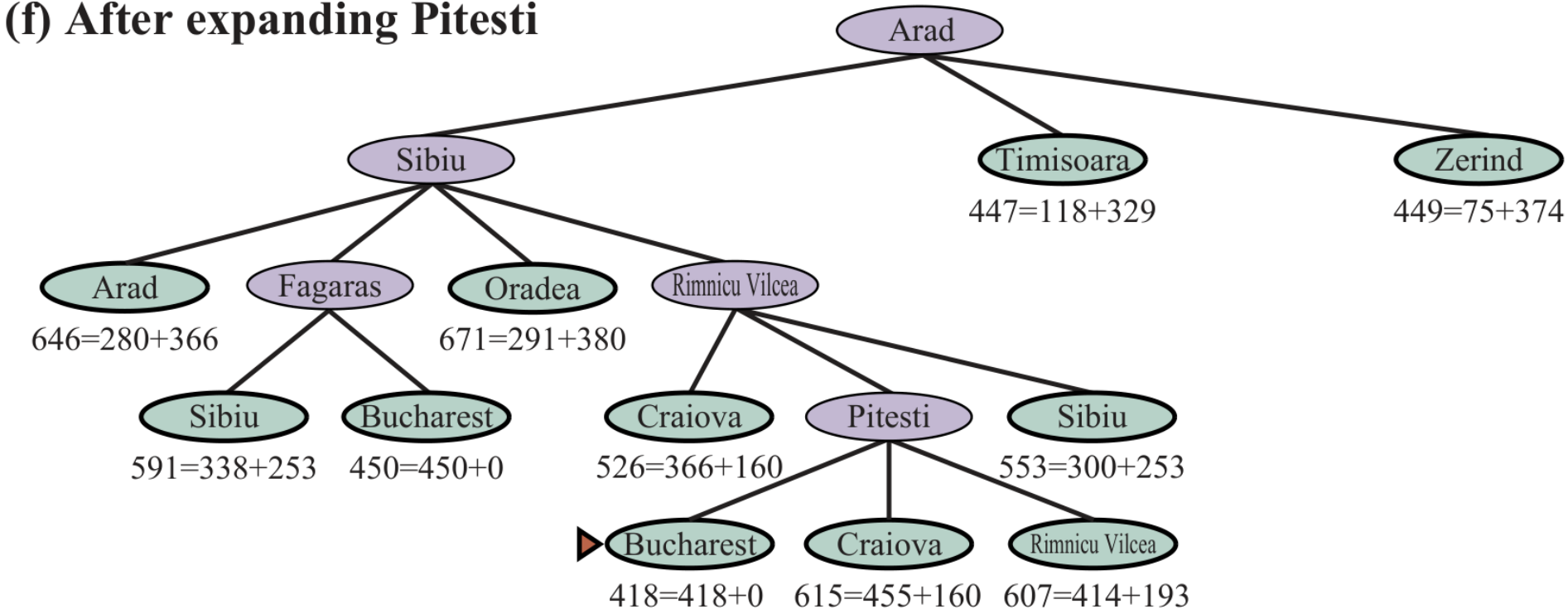
Arad to Bucharest: A* Algorithm



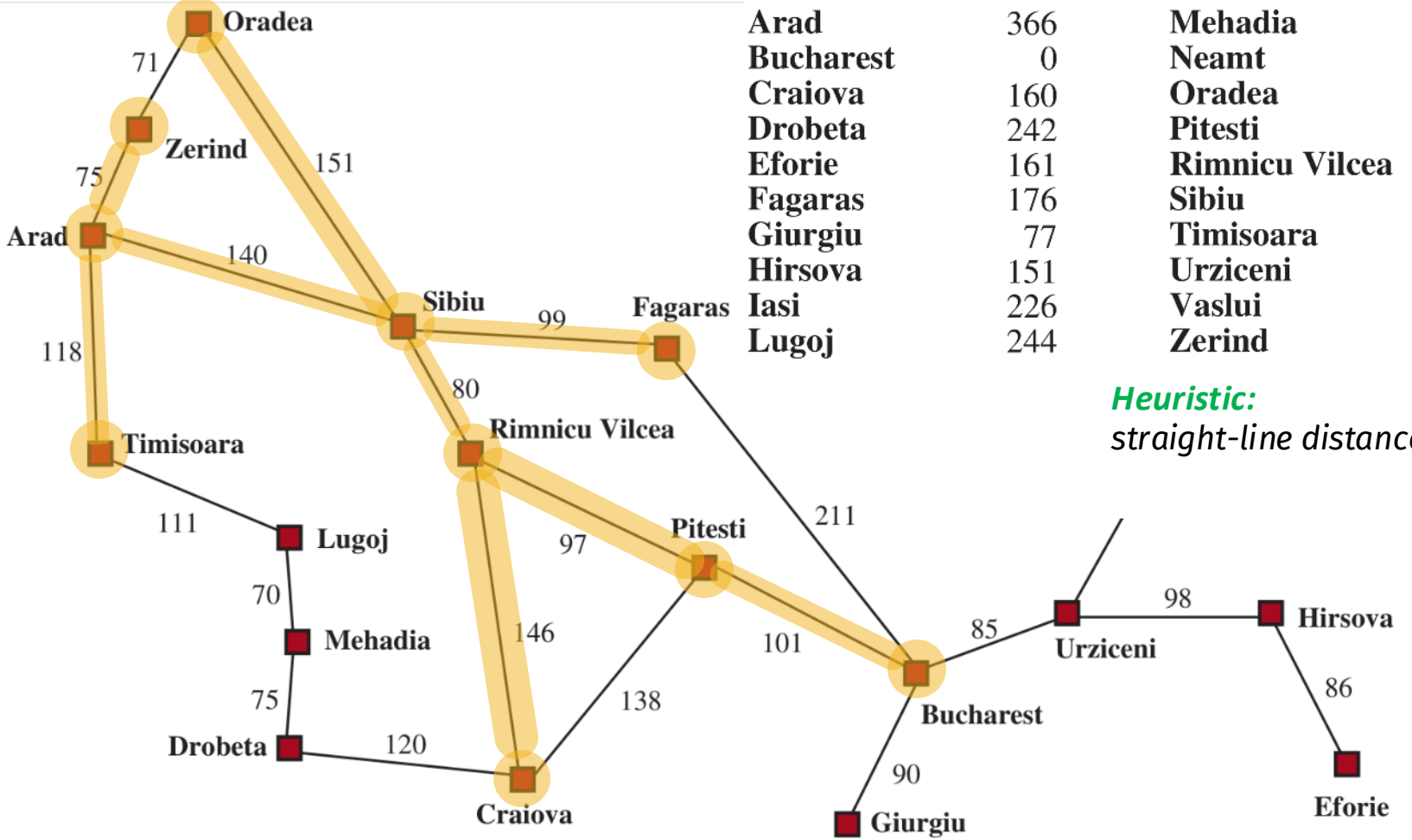
Arad to Bucharest: A* Algorithm



Arad to Bucharest: A* Algorithm



Arad to Bucharest: A* Algorithm



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

Heuristic:
straight-line distance

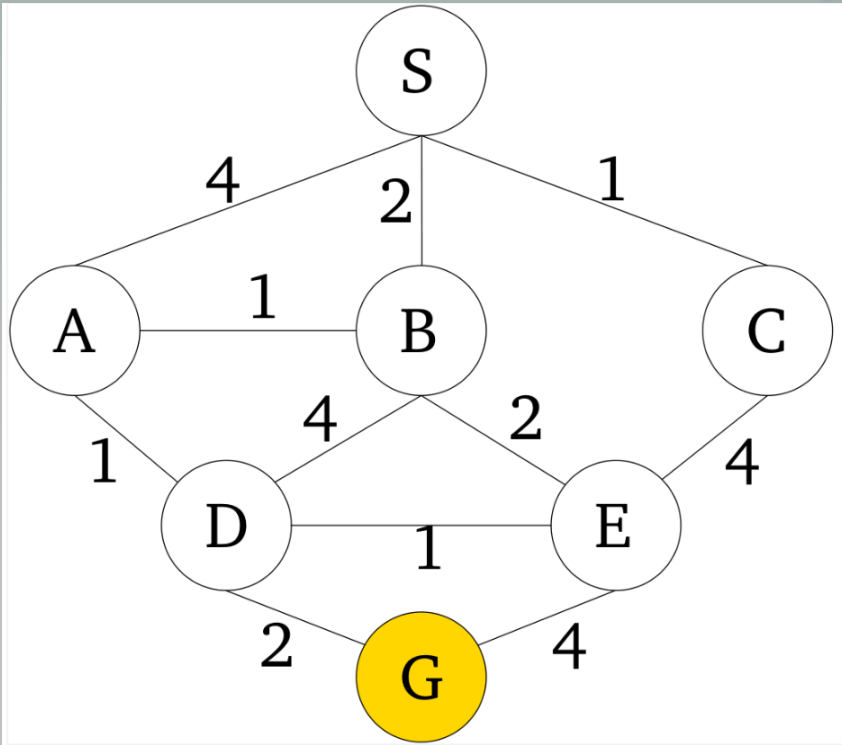
Arad to Bucharest: A* Algorithm

Exercise: A* Algorithm

Gunakan algoritma A* untuk mencari path paling optimal (cost terkecil) dari S ke G.

Saat expand, jika ada *f(n)* yang sama, dipilih secara lexicographic.

Pada setiap langkah, tuliskan nilai *f(n)=g(n)+h(n)* untuk semua node lain yang masih berada di frontier
Apakah *h( )* admissible?



h()	
S	6
A	3
B	5
C	4
D	1
E	4
G	1

#0
S(0+6 = 6)

#1: expand S, cost 0
A(4+3 = 7)
B(2+5 = 7)
C(1+4 = 5)

#2: expand C, cost 1
A(4+3 = 7)
B(2+5 = 7)
E(5+4 = 9)

#3: expand A, cost 4
B(2+5 = 7)
E(5+4 = 9)
D(5+1 = 6)
new B (5+5=10), lebih besar, abaikan

#4: expand D, cost 5
B(2+5 = 7)
E(5+4 = 9)
new B (9+5 = 13), lebih besar, abaikan
new E(6+4 = 10), lebih besar, abaikan
G(7+1=8)

#5: expand B, cost 2
E(5+4 = 9)
G(7+1=8)
new E(4+4 = 8), lebih kecil, update

#6: expand E, cost 4
G(7+1=8)
new G(8+1 = 9), lebih besar, abaikan

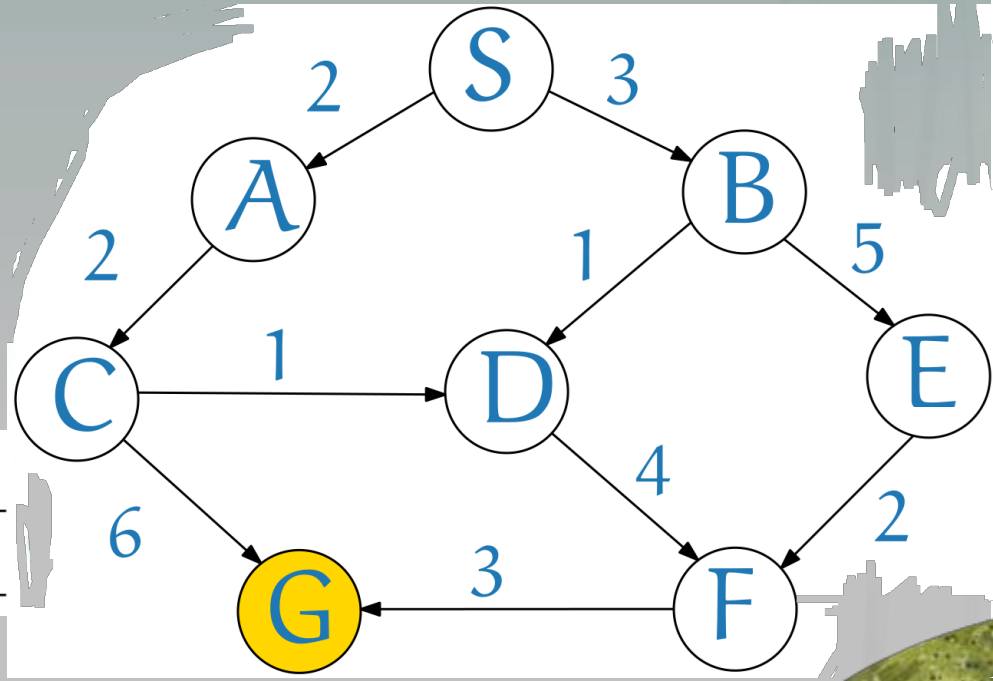
#7: mencapai state G, cost 7, selesai
h() tidak admissible, karena
-h(G) > 0
-h(B) = 5, lebih besar dari minimum path sebenarnya B-A-D-G = 4

Exercise: Heuristic Function

Berapakah rentang nilai *h(C)* dan *h(F)* agar *h()* menjadi fungsi heuristik yang *admissible* dan *consistent*?
Jelaskan jawaban anda!

Nilai dalam rentang selalu positif.
Misal: $0 \leqslant h(C) \leqslant 1000$.

	S	A	B	C	D	E	F	G
h()	9	7	6	??	5	2	??	0



Admissible: C->G 0 <=h(C) <= 6
consistent:
h(A) <= d(A-C) + h(C) --> 7 <= 2 + h(C) --> 5 <= h(C)
h(C) <= d(G-C) + h(G) --> h(C) <= 6 + 0 --> h(C) <= 6
h(C) <= d(D-C) + h(D) --> h(C) <= 1 + 5 --> h(B) <= 6
5 <= h(C) <= 6
Admissible: F->G 0 <=h(F) <= 3
consistent:
h(D) <= d(D-F) + h(F) --> 5 <= 4 + h(F) --> 1 <= h(F)
h(E) <= d(E-F) + h(F) --> 2 <= 2 + h(F) --> 0 <= h(F)
h(F) <= d(G-F) + h(G) --> h(F) <= 3 + 0 --> h(F) <= 3
1 <= h(F) <= 3