Informed Search

Oren Etzioni
CSE 473
University of Washington

Best-first Search

Generalization of breadth first search
Priority queue of nodes to be explored
Cost function f(n) applied to each node

```
Add initial state to priority queue
While queue not empty
    Node = head(queue)
    If goal?(node) then return node
    Add children of node to queue
```

Old Friends

- Breadth first = best first
  With f(n) = depth(n)

- Dijkstra's Algorithm = best first
  With f(n) = g(n)
  Where g(n) = sum of edge costs from start to n
  Space bound (stores all generated nodes)

A* Search

- Hart, Nilsson & Rafael 1968

  Best first search with f(n) = g(n) + h(n)

  Where g(n) = sum of edge costs from start to n

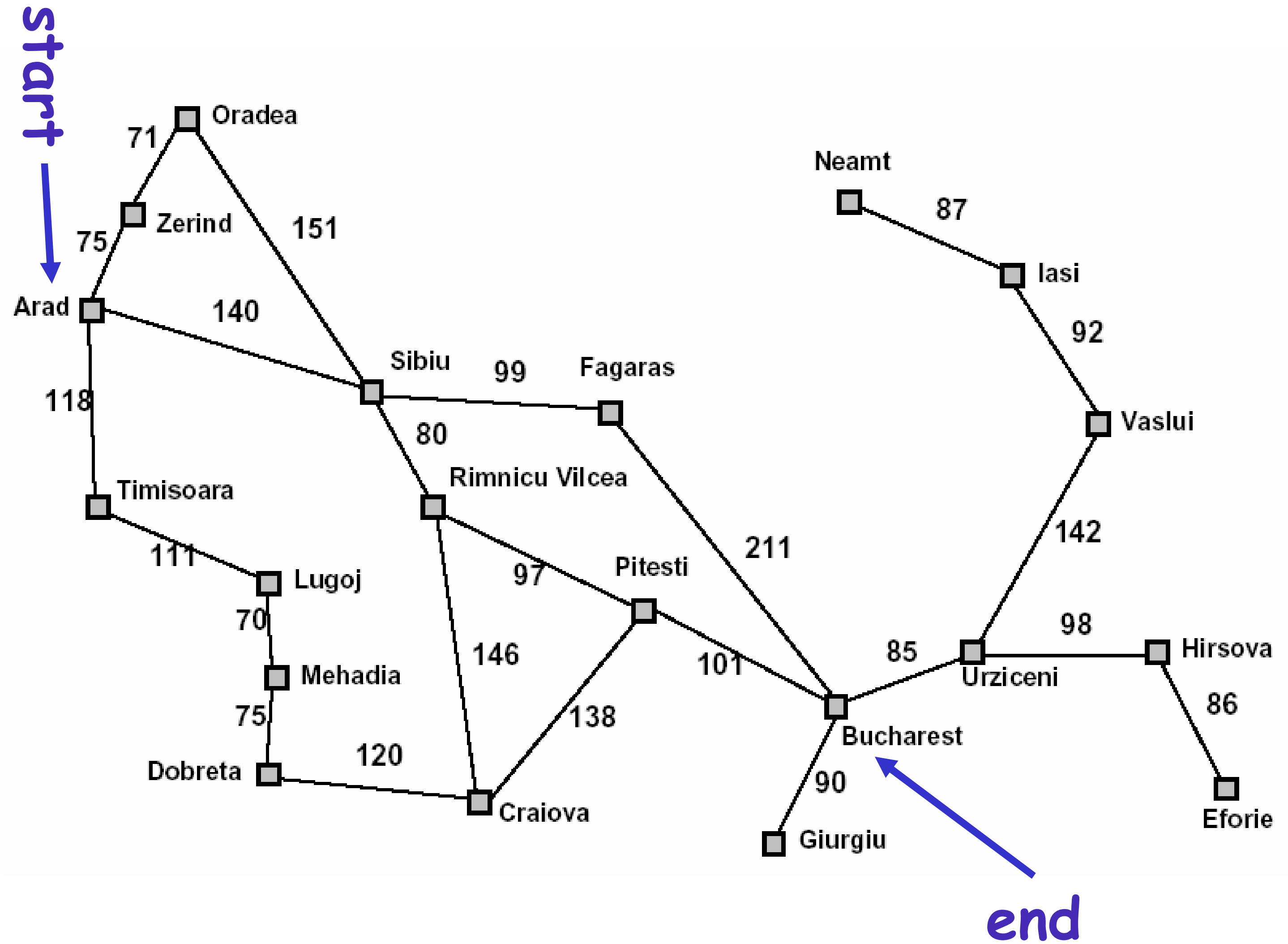
  And h(n) = estimate of lowest cost path n-->goal

  If h(n) is **admissible** then search will find optimal

  { Underestimates cost of any solution which can reached from node

Space bound since must maintain queue

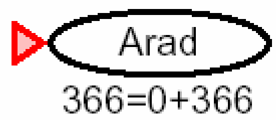
European Example



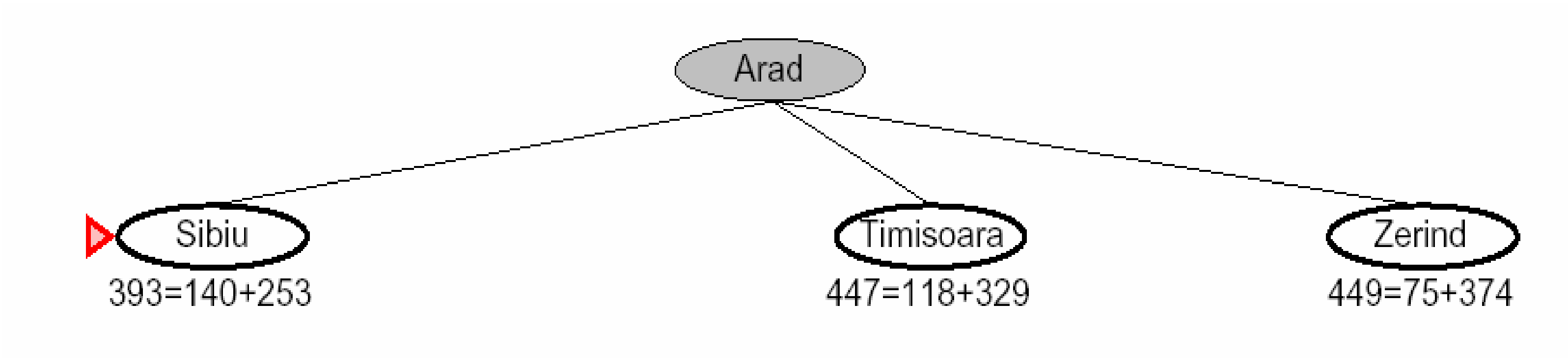
Straight−line distance
to Bucharest

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* Example

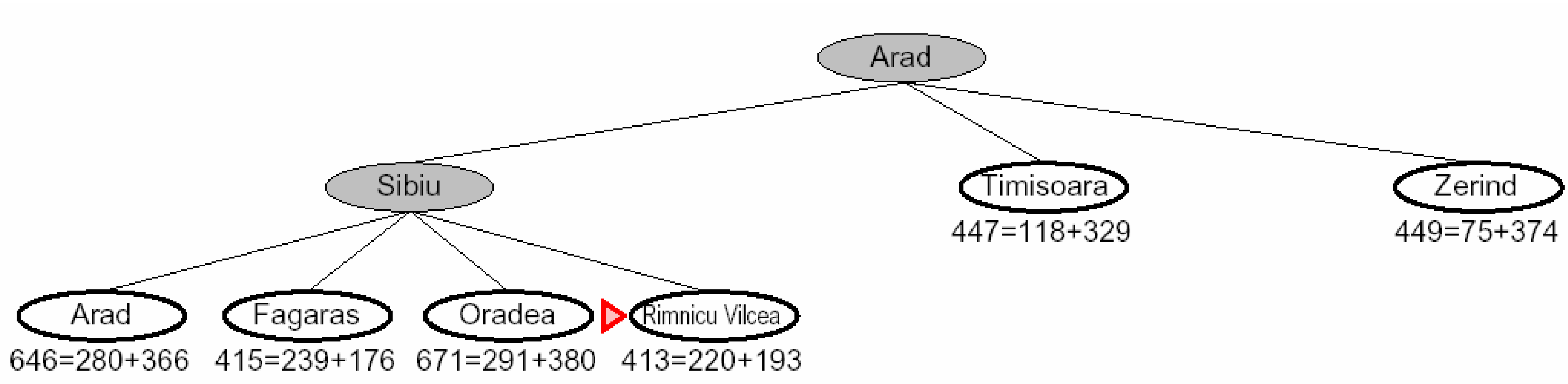


A* Example



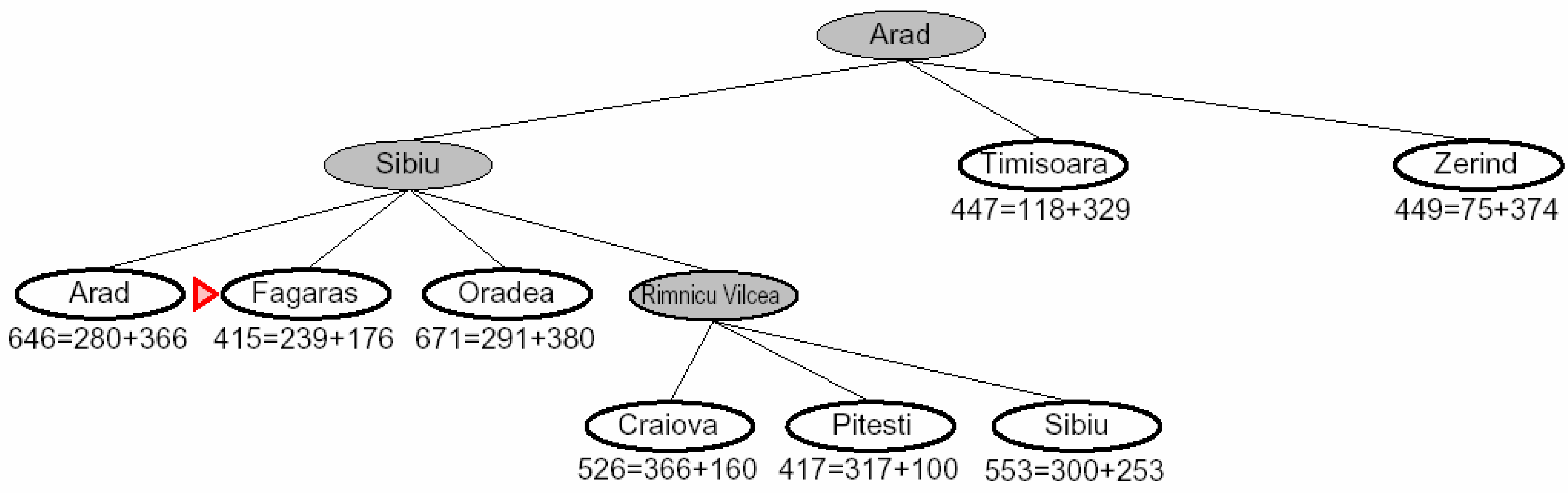
Arad
Sibiu
393=140+253
Timisoara
447=118+329
Zerind
449=75+374

A* Example

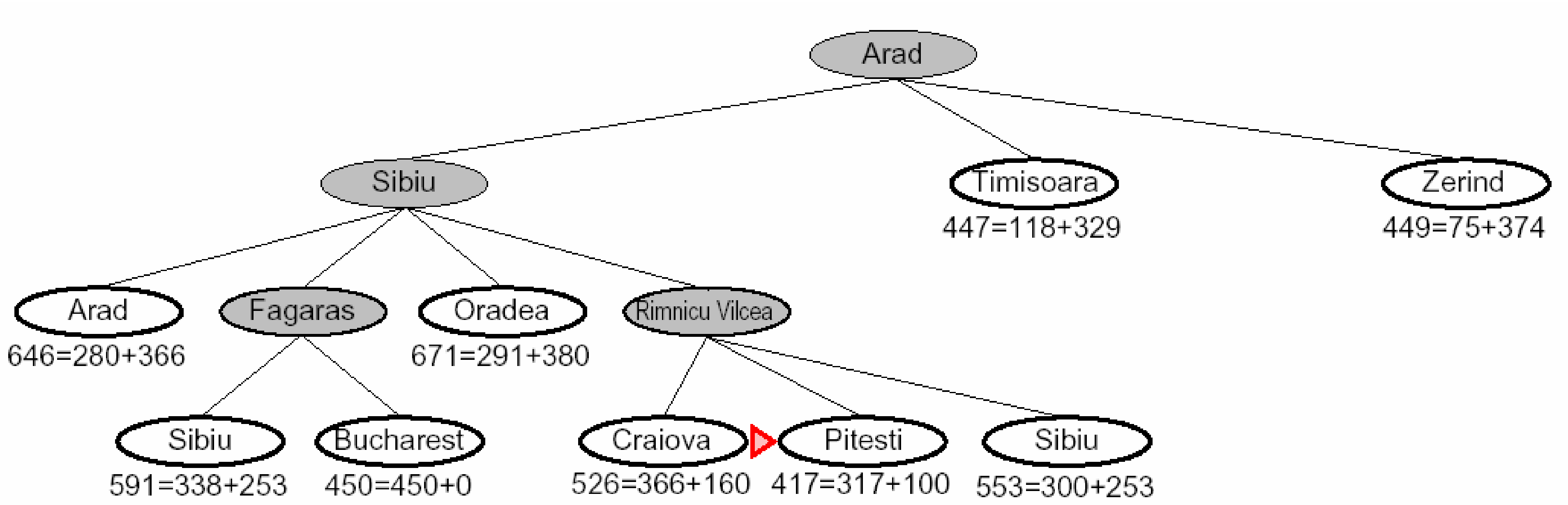


Arad
Sibiu
Timisoara
447=118+329
Zerind
449=75+374
Arad
646=280+366
Fagaras
415=239+176
Oradea
671=291+380
Rimnicu Vilcea
413=220+193

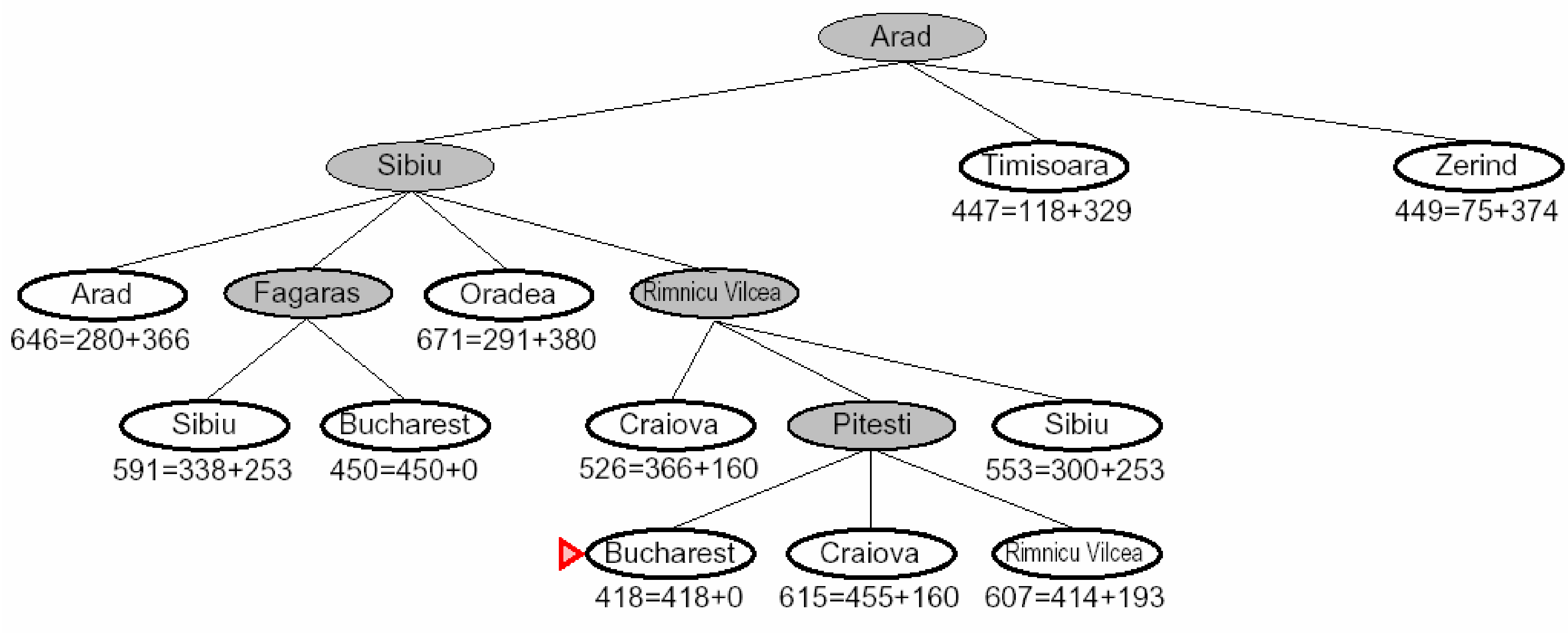
A* Example


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

A* Example

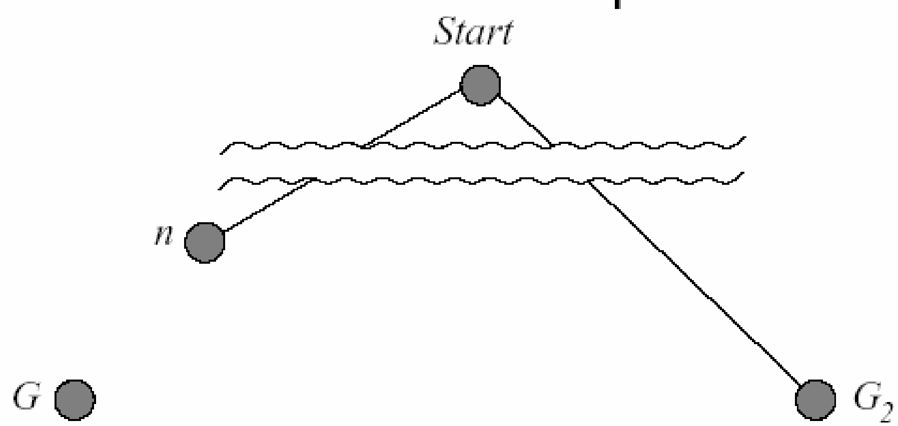

Arad
Sibiu
Timisoara
447=118+329
Zerind
449=75+374
Arad
646=280+366
Fagaras
Oradea
671=291+380
Rimnicu Vilcea
Sibiu
591=338+253
Bucharest
450=450+0
Craiova
526=366+160
Pitesti
417=317+100
Sibiu
553=300+253

A* Example


Arad
Sibiu
Timisoara
447=118+329
Zerind
449=75+374
Arad
646=280+366
Fagaras
Oradea
671=291+380
Rimnicu Vilcea
Sibiu
591=338+253
Bucharest
450=450+0
Craiova
526=366+160
Pitesti
Sibiu
553=300+253
Bucharest
418=418+0
Craiova
615=455+160
Rimnicu Vilcea
607=414+193

Optimality of A*

Suppose some suboptimal goal G_2 has been generated and is in the queue. Let n be an unexpanded node on a shortest path to an optimal goal G_1.



$$
\begin{aligned}
f(G_2) &= g(G_2) && \text{since } h(G_2) = 0 \\
&> g(G_1) && \text{since } G_2 \text{ is suboptimal} \\
&\geq f(n) && \text{since } h \text{ is admissible}
\end{aligned}
$$

Since $f(G_2) > f(n)$, A* will never select G_2 for expansion

A* Summary

- Pros
- Cons

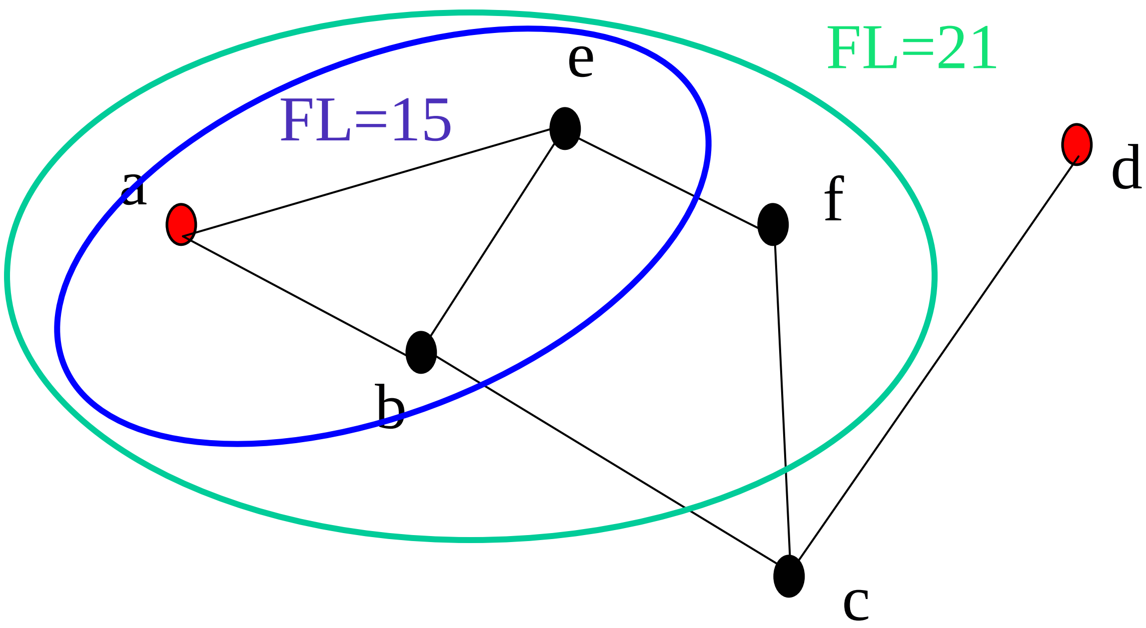
Iterative-Deepening A*

- Like iterative-deepening depth-first, but...
- Depth bound modified to be an **f-limit**

  Start with limit = h(start)

  Prune any node if f(node) > f-limit

  Next f-limit=min-cost of any node pruned



Depth-First Branch & Bound

- Single DF search
  → uses linear space
- Keep track of best solution so far
- If $f(n) = g(n)+h(n) \geq cost(best\text{-}soln)$
  Then prune n

- Requires
  Finite search tree, or
  Good upper bound on solution cost

Beam Search

- Idea
  - Best first but only keep N best items on priority queue
- Evaluation
  - Complete?

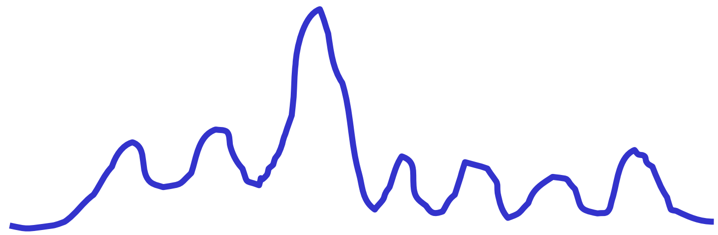
  - Time Complexity?

  - Space Complexity?

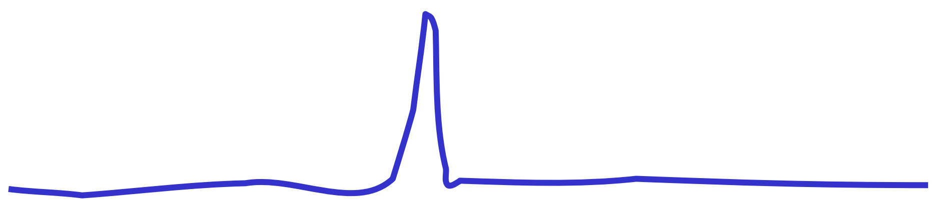
Hill Climbing

"Gradient ascent"

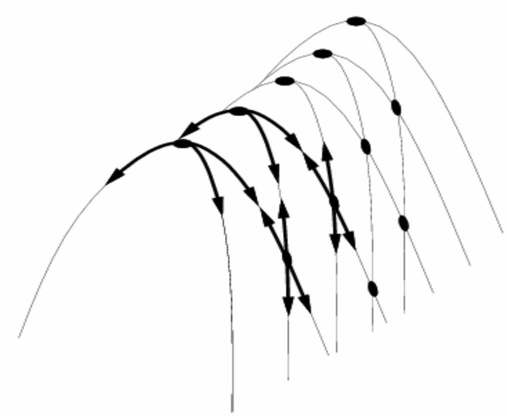
- Idea
  - Always choose best child; no backtracking
  - Beam search with |queue| = 1
- Problems?
  - Local maxima



  - Plateaus



  - Diagonal ridges



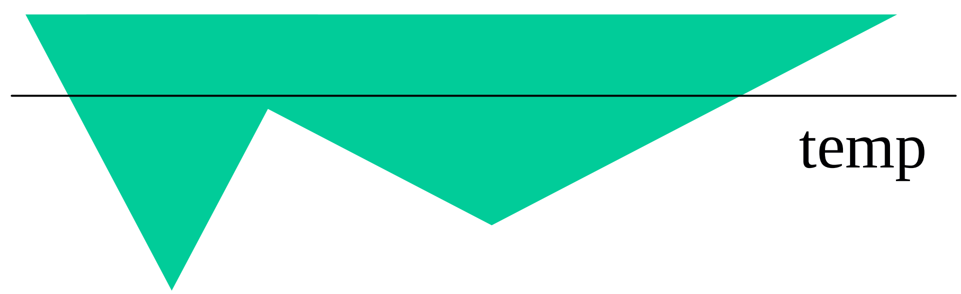
Randomizing Hill Climbing

- Randomly disobeying heuristic
- Random restarts

- [Add something on heavy tailed distribution]

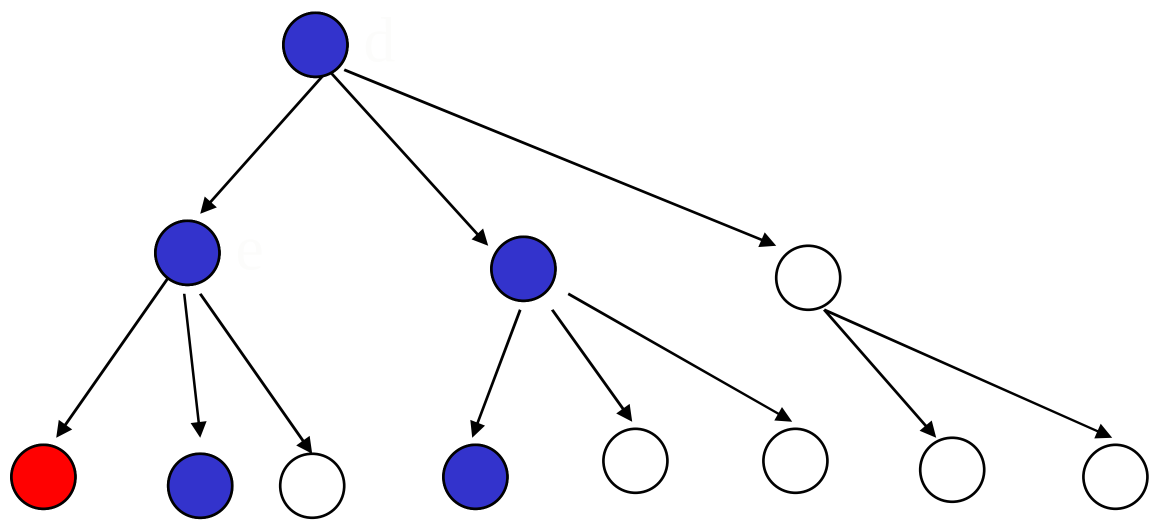
Simulated Annealing

- Objective: avoid local minima
- Technique:
  - For the most part use hill climbing
  - When no improvement possible
    - Choose random neighbor
    - Let Δ be the decrease in quality
    - Move to neighbor with probability $e^{-\Delta-/T}$
  - Reduce "temperature" (T) over time
- Pros & cons
  - Optimal?
  - If T decreased slowly enough, ***will*** reach optimal state
- Widely used
- See also
  - WalkSAT



Limited Discrepancy Search

- Discrepancy bound indicates how often to violate heuristic
- Iteratively increase...



Assume that heuristic says go ***left***

Genetic Algorithms

- Start with random population
  - Representation serialized
  - States are ranked with "fitness function"
- Produce new generation
  - Select random pair(s):
    - probability ~ fitness
  - Randomly choose "crossover point"
    - Offspring mix halves
  - Randomly mutate bits

Crossover **Mutation**

174629844710	→	174611094281	→	164611094281
776511094281		776529844710		776029844210

Properties

- Importance of careful reprsentation