Artificial Intelligence

[week #3]

Heuristic Search

Hilmy. A. T

hilmi.tawakal@gmail.com

State space search

- **Uninformed search:** algorithms that are given no information about the problem other than its definition.
  - Breadth-first search
  - Depth-first search
  - Uniform-cost search
  - Depth-limited search
  - Iterative deepening depth-first search
- **Informed / heuristic search:** uses problem-specific knowledge beyond the definition of the problem itself
  - Greedy best first search
  - A*

Search for sollution

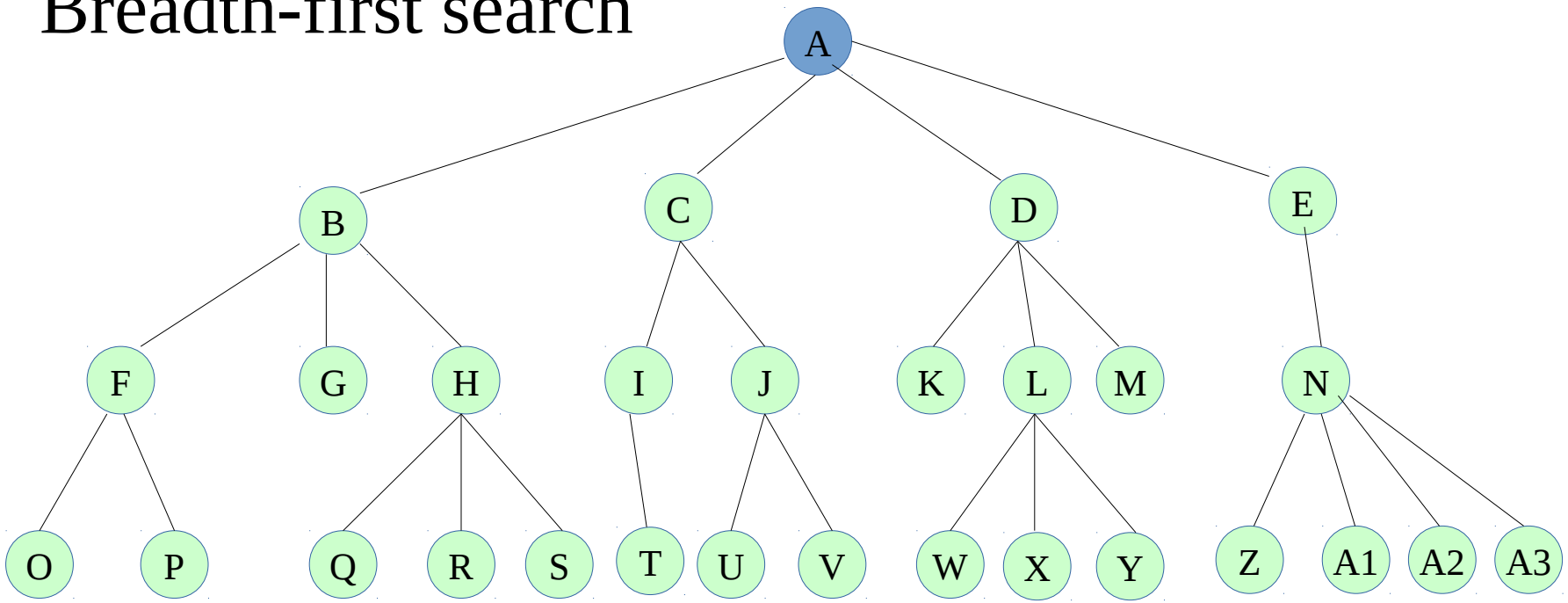
- Expand node with smallest ***f(n)*** in the *fringle*
- *Fringle* = unexpanded node
- *f(n)* → evaluation function
- *g(n)* → cost to reach n from current node
- *h(n)* → prediction how good the node is
  - straight line distance
  - manhattan distance
  - etc

State space search

- **Uninformed search:**
  - Breadth-first search → f(n) = FIFO
  - Depth-first search → f(n) = LIFO
  - Uniform-cost search → f(n) = g(n)
  - Depth-limited search → f(n) = LIFO ; d < threshold
  - Iterative deepening → f(n) = LIFO ; d < t; 0 < t < X
- **Informed / heuristic search:**
  - Greedy best first search → f(n) = h(n)
  - A* → f(n) = g(n) + h(n)

Uninformed search

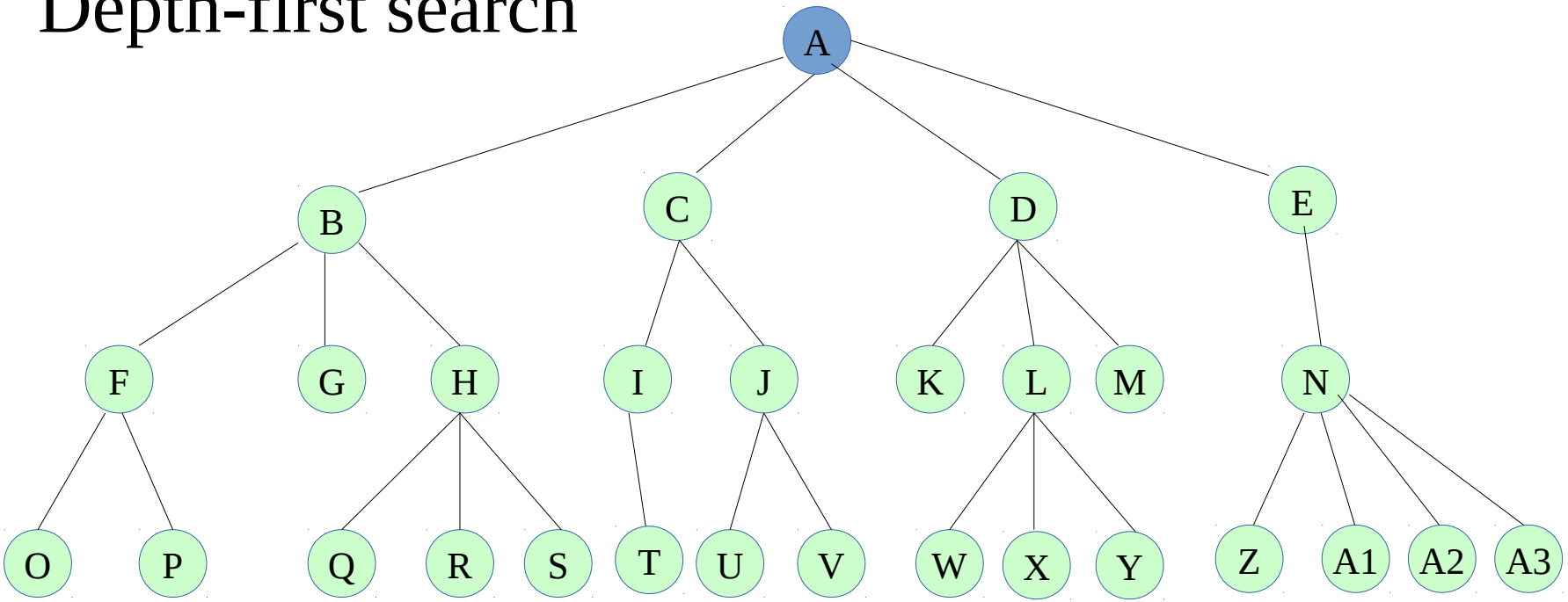
Breadth-first search



A-B-C-D-E-F-G-H-I-J-K- ...

Uninformed search

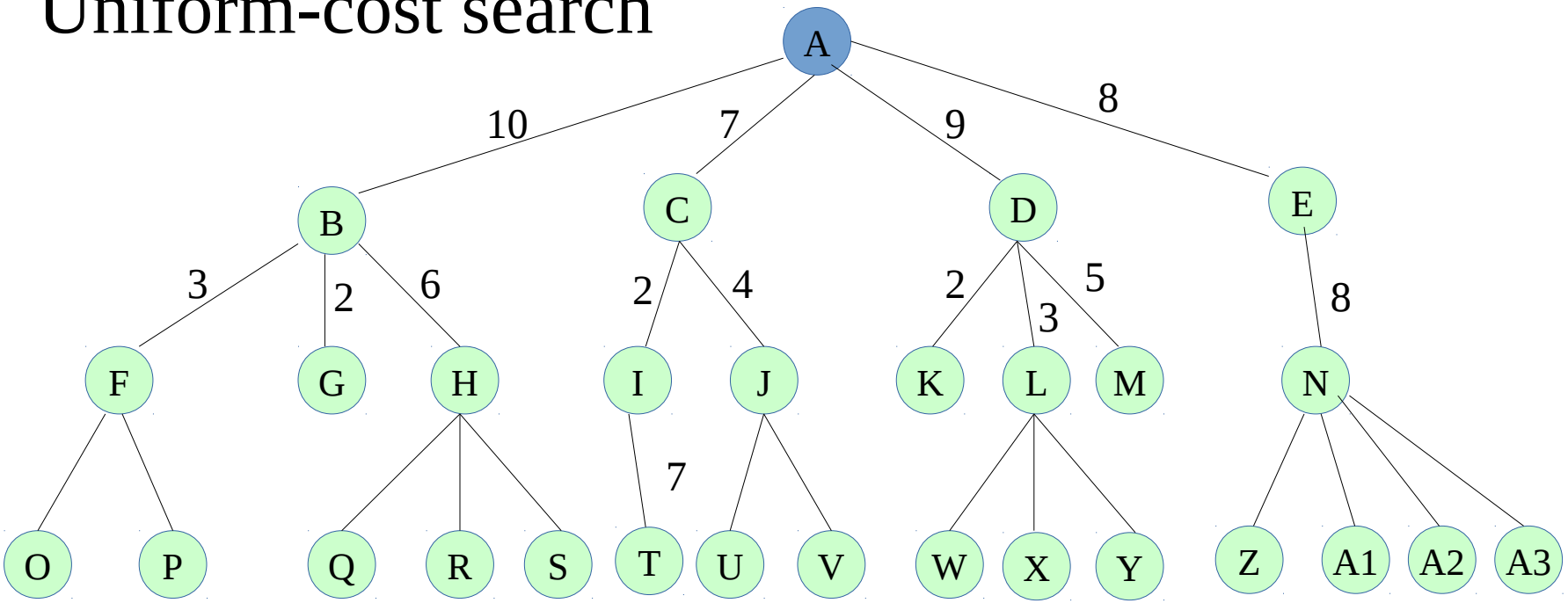
Depth-first search



A-B-F-O-P-G-H-Q-R-S-C-I-T-J- ...

Uninformed search

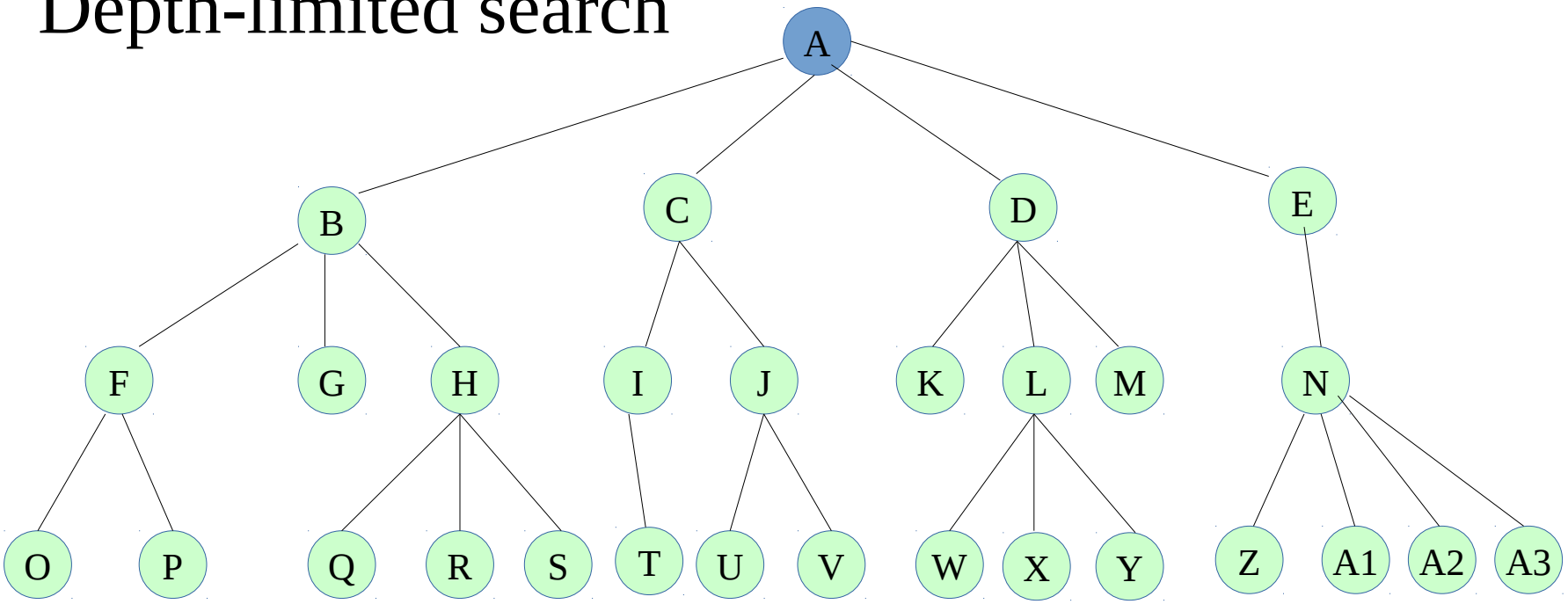
Uniform-cost search



A:0,C:7,E:8,I:9,D:9,B:10,K:11,J:11,G:12,L:12, ...

Uninformed search

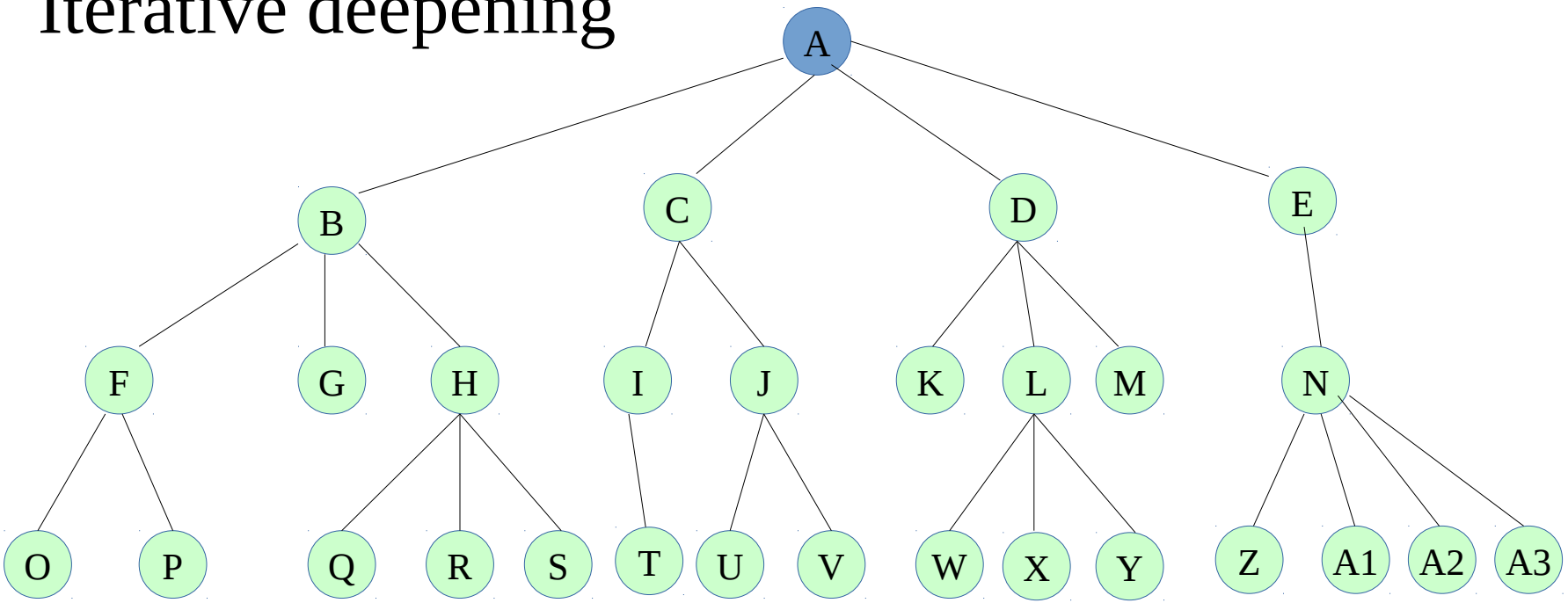
Depth-limited search



Limit:2 => A-B-F-G-H-C-I-J-D-K-L-M-E-N

Uninformed search

Iterative deepening

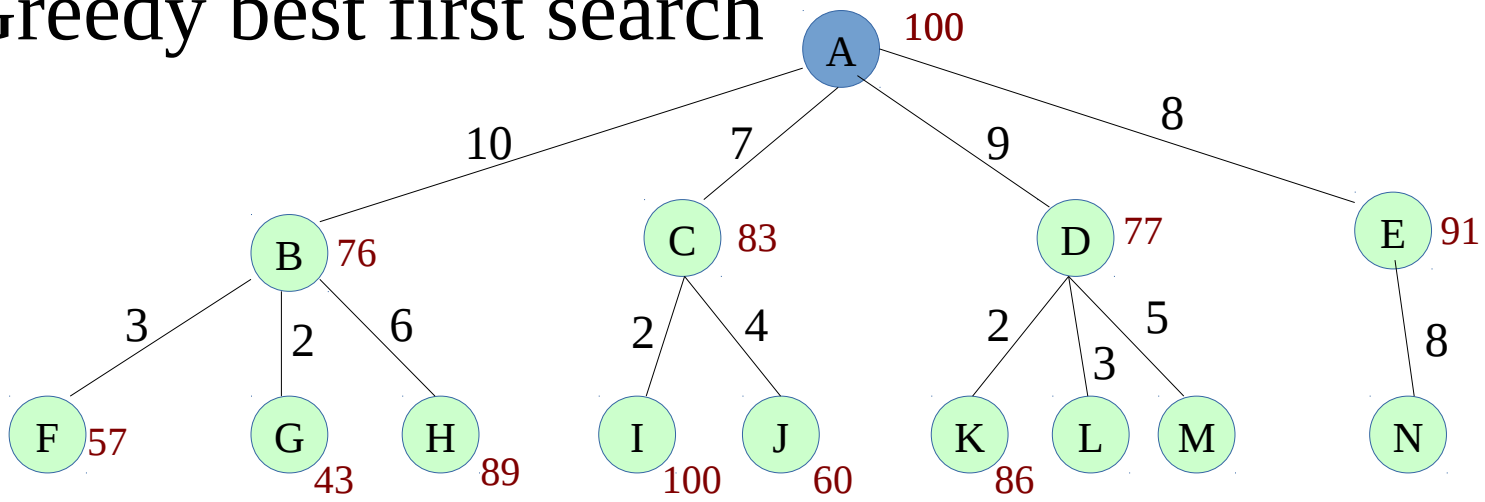


Iterasi #1 → Limit:1 => A-B-C-D-E

Iterasi #2 → Limit:3 => A-B-F-O-P-G-H-Q-R-S-C-I-T- . . .

Heuristic search

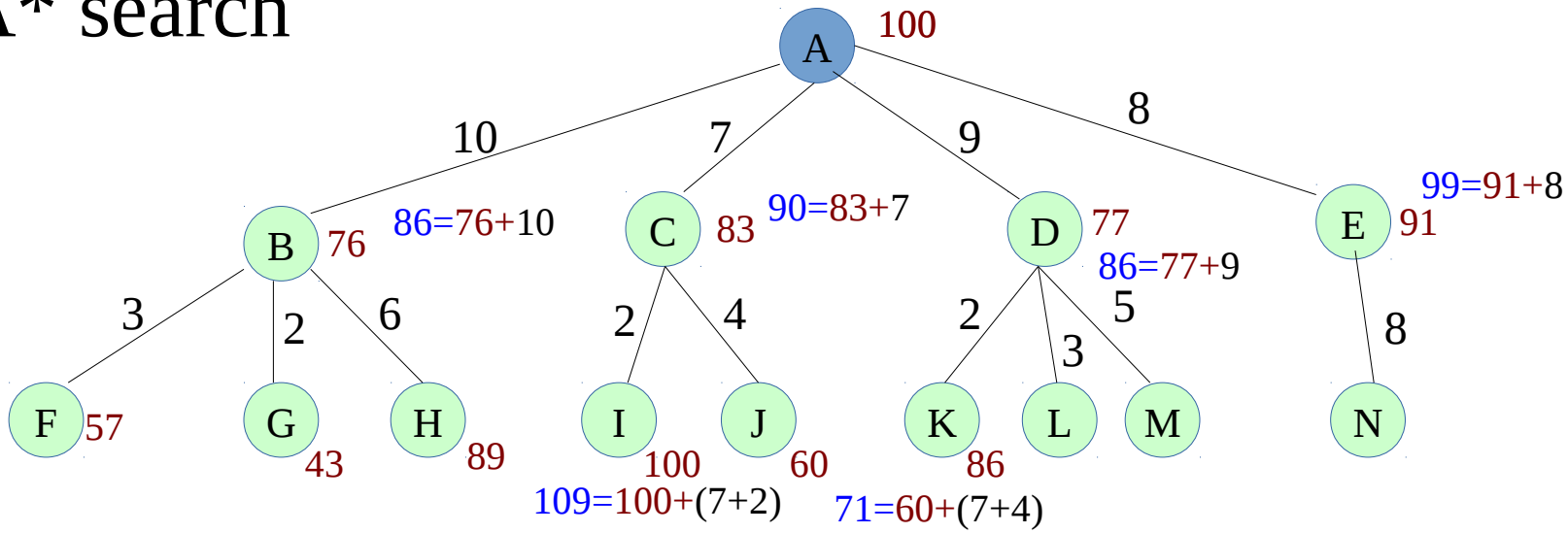
Greedy best first search



- f(n) = h(n)
- B:76, G:43, …
- Problem: if goal = I ? → infinite loop

Heuristic search

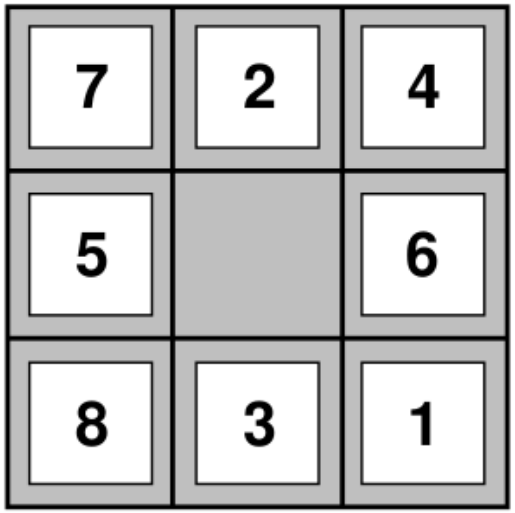
A* search



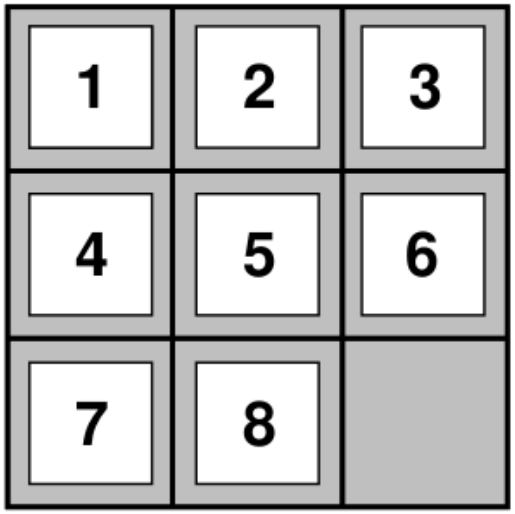
- f(n) = g(n) + h(n)
- Similar with uniform-cost search, uniform-cost search f(n)=g(n)

A:100=0+100, C:90=83+7, J:71=60+(7+4)

Designing Heuristic



Start State



Goal State

h1(n)= sum of incorrect position

→ h1(s) = 6

h2(n)=sum of distance from correct position

→ h1(s) = 4+0+3+3+1+0+2+1 = 14

Designing heuristic

- If h1(n) and h2(n) admissible. Which one is better?
- Compare the number of expanded node:

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

- if h2(n) ≥ h1(n) for all n, h2 is dominating h1 and better for search
- The bigger h(n)→ closer to h∗(n), more UN-expanded node (pruned) → more efficient!