

GREEDY BEST-FIRST SEARCH

CSE 511A: Introduction to Artificial Intelligence

Some content and images are from slides created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley.
All CS188 materials are available at <http://ai.berkeley.edu>.

1

UNINFORMED SEARCH

Generic uninformed search pseudo-code:

- (1) Start with a tree that contains only the start state
- (2) Pick an unexpanded fringe node n
- (3) If fringe node n represents a goal state, then stop
- (4) Expand fringe node n^*
- (5) Go to (2)

*generate neighboring nodes that aren't ancestors

2

BREADTH-FIRST SEARCH

Breadth-first search pseudo-code:

- (1) Start with a tree that contains only the start state
- (2) Pick an unexpanded fringe node n **with the smallest depth**
- (3) If fringe node n represents a goal state, then stop
- (4) Expand fringe node n^*
- (5) Go to (2)

*generate neighboring nodes that aren't ancestors

3

DEPTH-FIRST SEARCH

Depth-first search pseudo-code:

- (1) Start with a tree that contains only the start state
- (2) Pick an unexpanded fringe node n **with the largest depth**
- (3) If fringe node n represents a goal state, then stop
- (4) Expand fringe node n^*
- (5) Go to (2)

*generate neighboring nodes that aren't ancestors

4

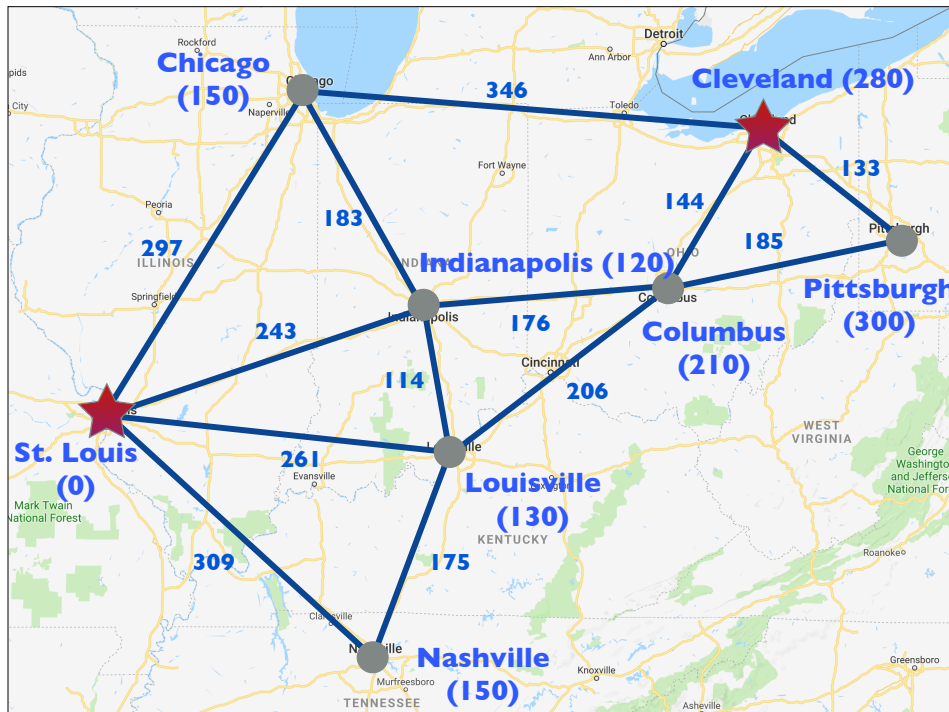
UNIFORM-COST SEARCH

Uniform-cost search pseudo-code:

- (1) Start with a tree that contains only the start state
- (2) Pick an unexpanded fringe node n **with the smallest $g(n)$**
- (3) If fringe node n represents a goal state, then stop
- (4) Expand fringe node n^*
- (5) Go to (2)

*generate neighboring nodes that aren't ancestors

5



7

GREEDY BEST-FIRST SEARCH

Greedy best-first search pseudo-code:

- (1) Start with a tree that contains only the start state
- (2) Pick an unexpanded fringe node n **with the smallest $h(n)$**
- (3) If fringe node n represents a goal state, then stop
- (4) Expand fringe node n^*
- (5) Go to (2)

*generate neighboring nodes that aren't ancestors

6

PROPERTIES

	BFS	UCS	Greedy
Correct the solution it finds is optimal	Yes if cost is uniform	Yes	No
Complete it terminates	Yes if b is finite	Yes	Yes if don't repeat states
Space Complexity max nodes in memory	$O(b^{d+1})$	$O(b^{\lceil C^*/\epsilon \rceil + 1})$	$O(b^m)$
Time Complexity max nodes generated	$O(b^{d+1})$	$O(b^{\lceil C^*/\epsilon \rceil + 1})$	$O(b^m)$

branching factor b
depth of the goal d
depth of tree m

cost of optimal solution C^*
cost increment ϵ

8

