

Artificial Intelligence For NLP Lesson-3

人工智能与自然语言处理课程组

2018.Nov. 18

Outline

- Search Based Decision Making
- Breath First Search, Depth First Search
- Best First Search, A* Search
- Decision tree pruning
- Branch and bound
- Eight Queen, The Missionaries and Cannibals Problem
- Assignment: Subway Route Planning

```

def search(graph, concat_func):
    seen = set()
    need_visited = ['1']

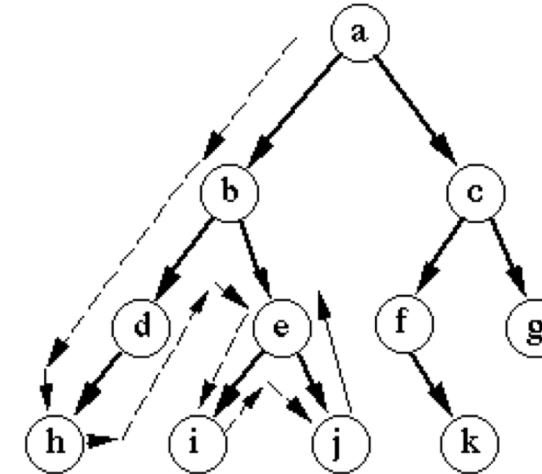
    while need_visited:
        node = need_visited.pop(0)
        if node in seen: continue
        print('  I am looking at : {}'.format(node))
        seen.add(node)
        new_discovered = graph[node]
        need_visited = concat_func(new_discovered, need_visited)

def treat_new_discover_more_important(new_discovered, need_visited):
    return new_discovered + need_visited

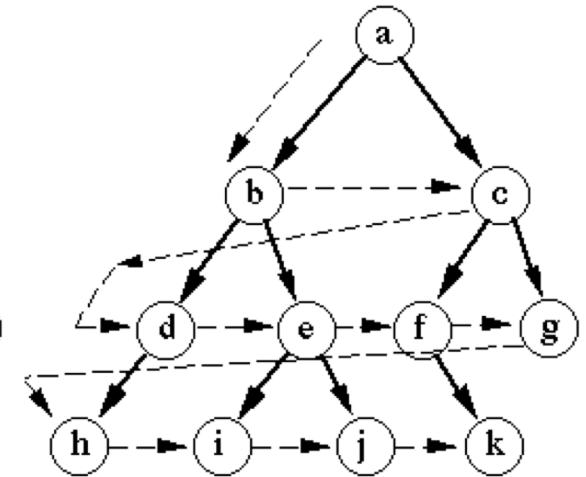
def treat_already_discovered_more_important(new_discovered, need_visited):
    return need_visited + new_discovered

dfs = partial(search, concat_func=treat_new_discover_more_important)
 bfs = partial(search, concat_func=treat_already_discovered_more_important)

```



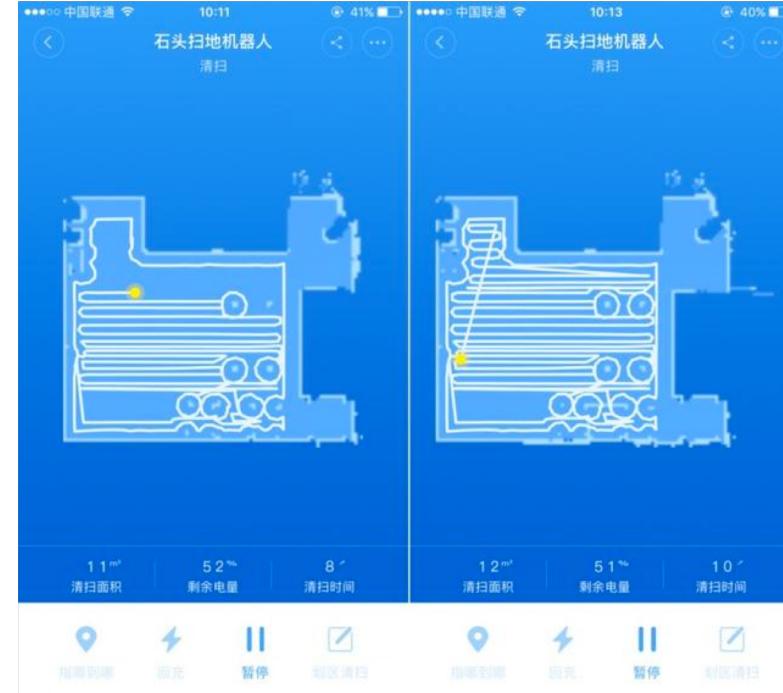
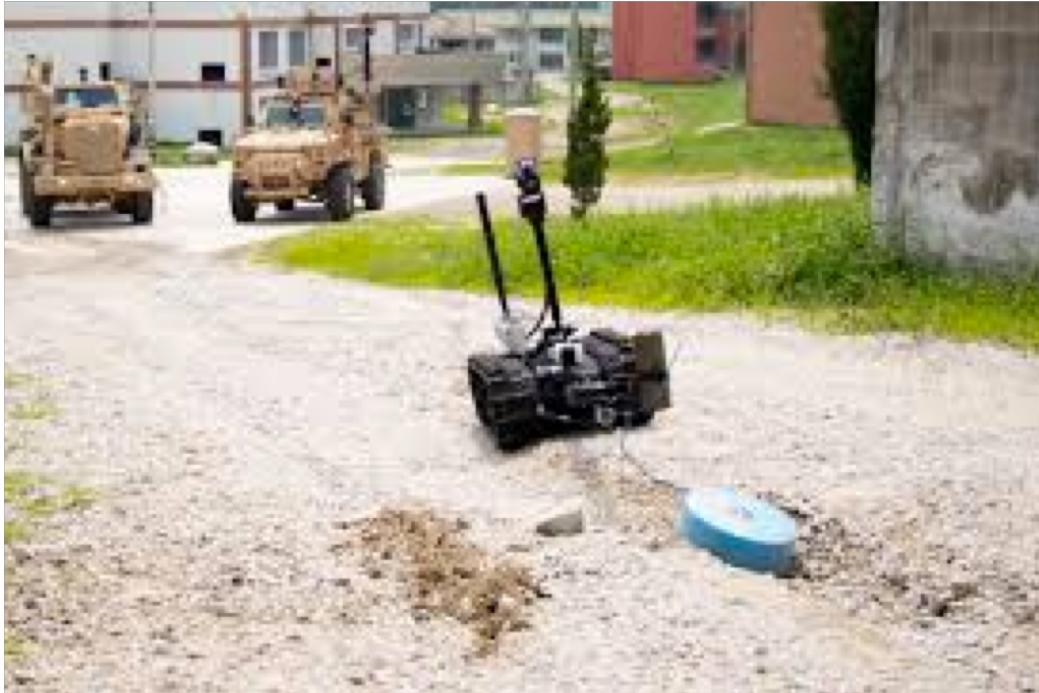
Depth-first search



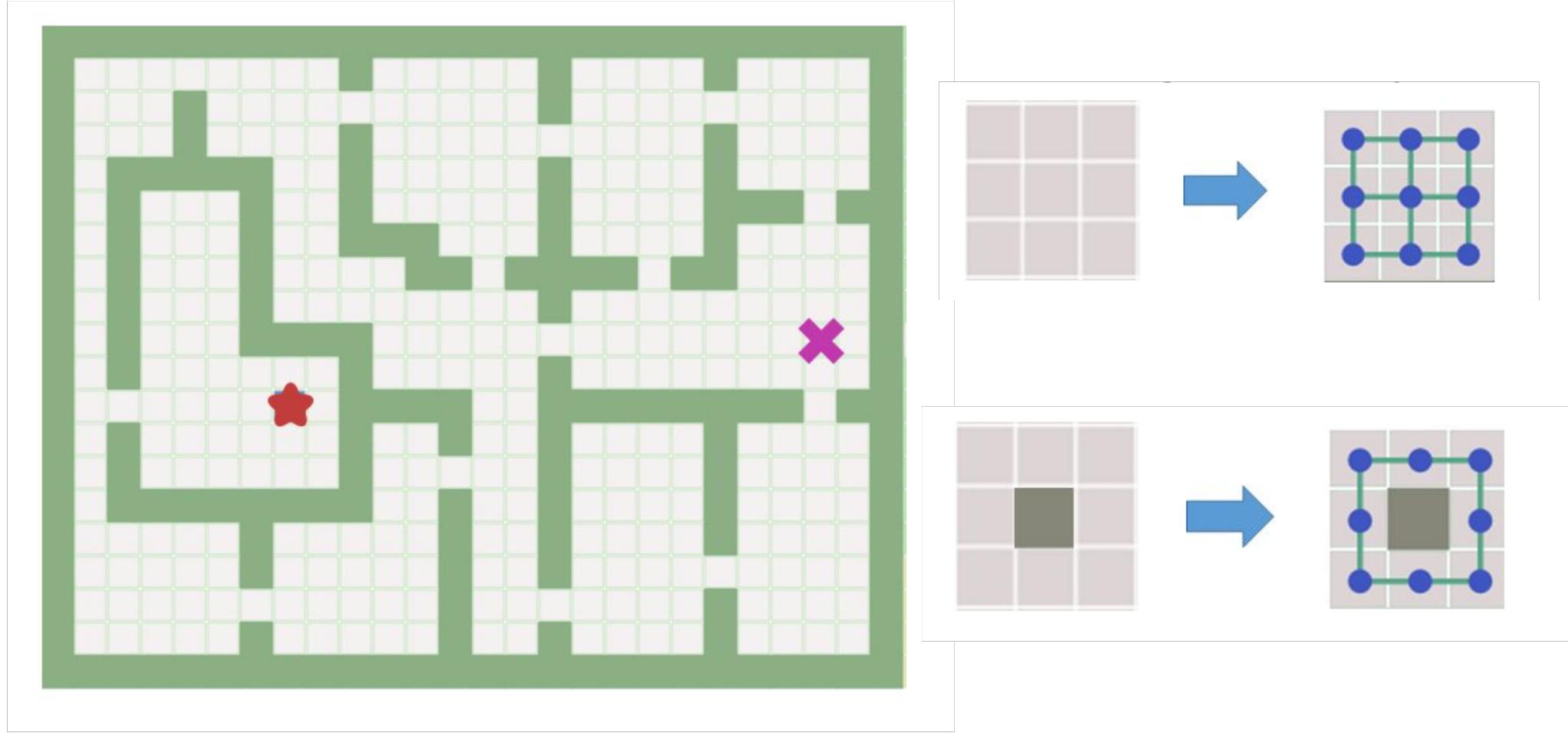
Breadth-first search

Review

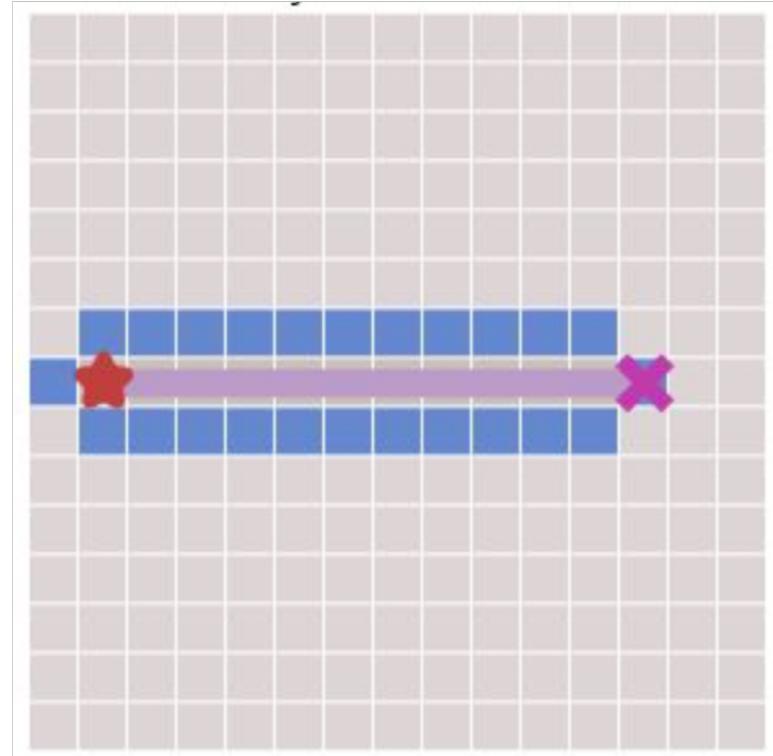
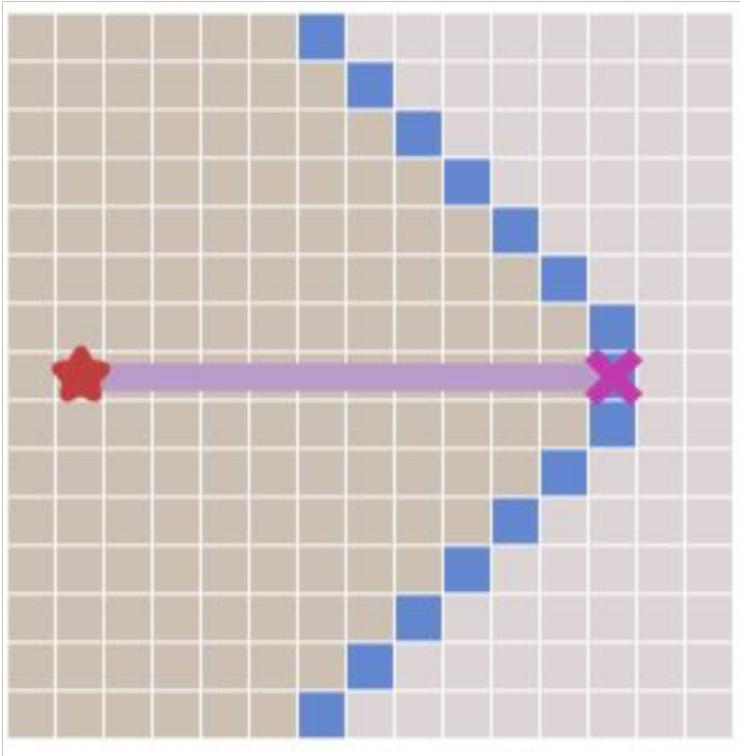
Applications



- The *start* state
- The *goal* state
- The *successors*
- The strategy that determines the order in which we search.



From Breadth First Search to Best First Search



A *heuristic* is an approximate measure of how close you are to the target

Activity

- What's the heuristic function of one problem?

- Map Routing



- Find the person-to-person connection in a social web.



A* search

At each iteration of its main loop, A* needs to determine which of its paths to extend. It does so based on the cost of the path and an estimate of the cost required to extend the path all the way to the goal. Specifically, A* selects the path that minimizes

$$f(n) = g(n) + h(n)$$

Like BFS, it finds the shortest path, and like Greedy Best First, it's fast.

Each iteration, A* chooses the node on the frontier which minimizes:

steps from source + approximate steps to target

From Search to DeepLearning

Decision Making and Dynamic Programming