Wang Xiyu

Problem Statement

fully observable $\wedge$ deterministic $\wedge$ static $\wedge$ discrete $\implies$ only need to observe once
To solve a prob using search:

- A goal or a set of goals
- a model of the enironment
- a search algorithm

goal formulation -> problem formulation -> search -> execute

1. goal formulation
2. problem formulation, eg. path finding
   - states: nodes representation invariant:: abstract states should correspond to concrete states
   - initial state: starting node
   - goal states/test: dest node
     Goal test: define the goal using a function is_goal
   - actions: move along an edge :: $|actions(state)| \leq (branching\_factor)$
   - transition model: $(curr\_state, action) \implies next\_state$
   - action cost function: see edges
3.

Search

Uninformed search

No information that could guide the seaech: mo clue how good a state is

```
create frontir,
...
```

frontier: queue: BFS

Depth limited search

limit the search to depth l
backtrack when the limit is hit.
time complexity: exponential to search depth
space complexity: size of the frontier

Iterative deeptening search

seatch with depth from 0 to inf
return soln when found. Both complete

Concept Proof

Solution

Easy