Introduction to Artificial Intelligence

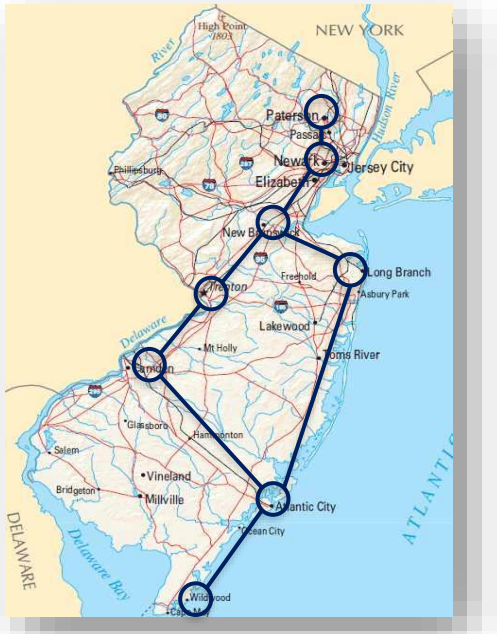
Course 16 :198 :440

Recitation 5:

Adversarial Search

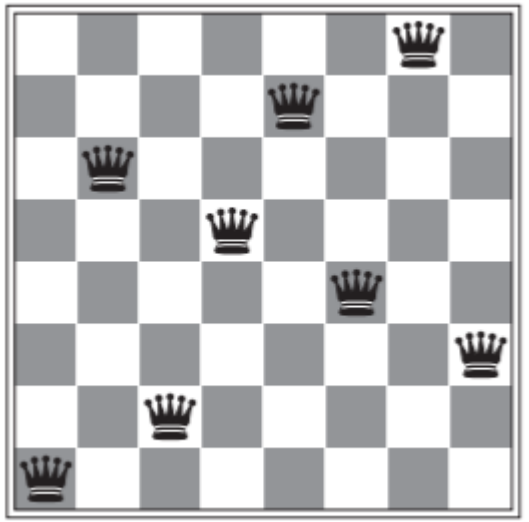
RUTGERS

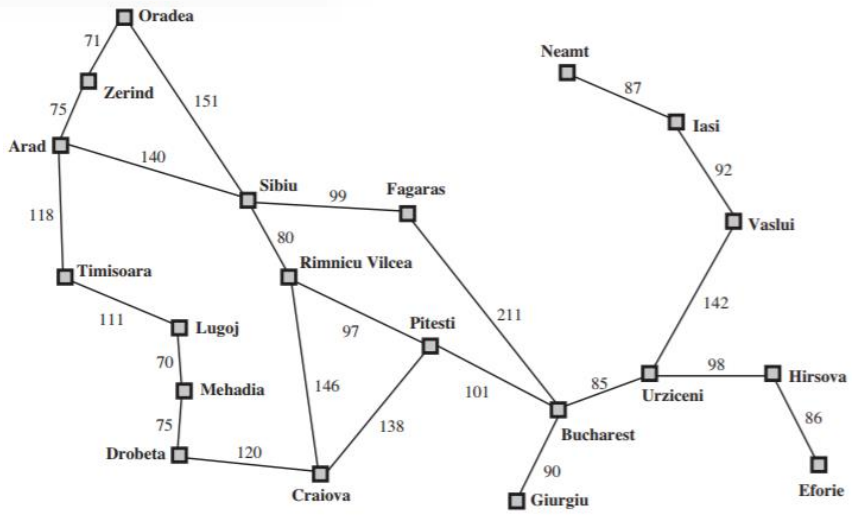
Adversarial Search

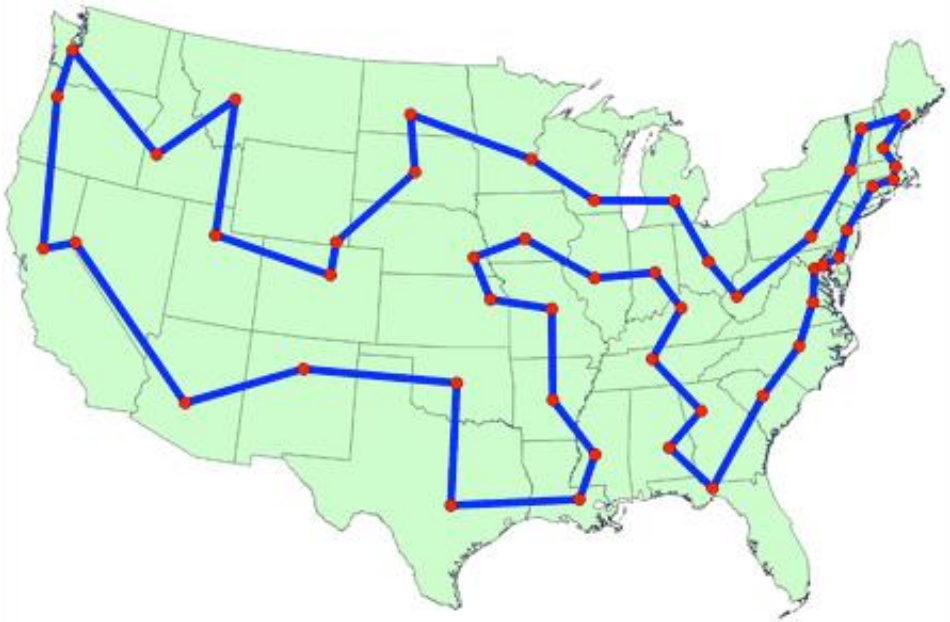


In a very general sense, we've considered searches in state spaces.

- Consider the four examples below, what were we trying to solve? And how was it done?





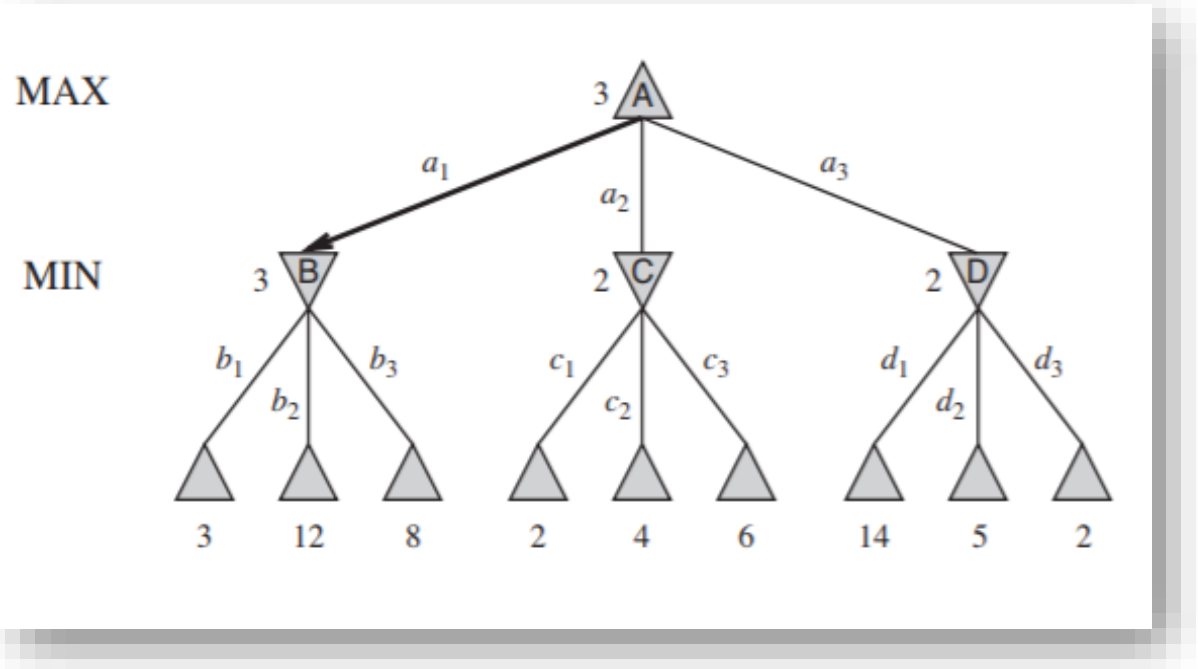


The Minimax algorithm

```
function MINIMAX-DECISION(state) returns an action
    return arg max_{a ∈ ACTIONS(s)} MIN-VALUE(RESULT(state, a))
```

```
function MAX-VALUE(state) returns a utility value
    if TERMINAL-TEST(state) then return UTILITY(state)
    v ← −∞
    for each a in ACTIONS(state) do
        v ← MAX(v, MIN-VALUE(RESULT(s, a)))
    return v
```

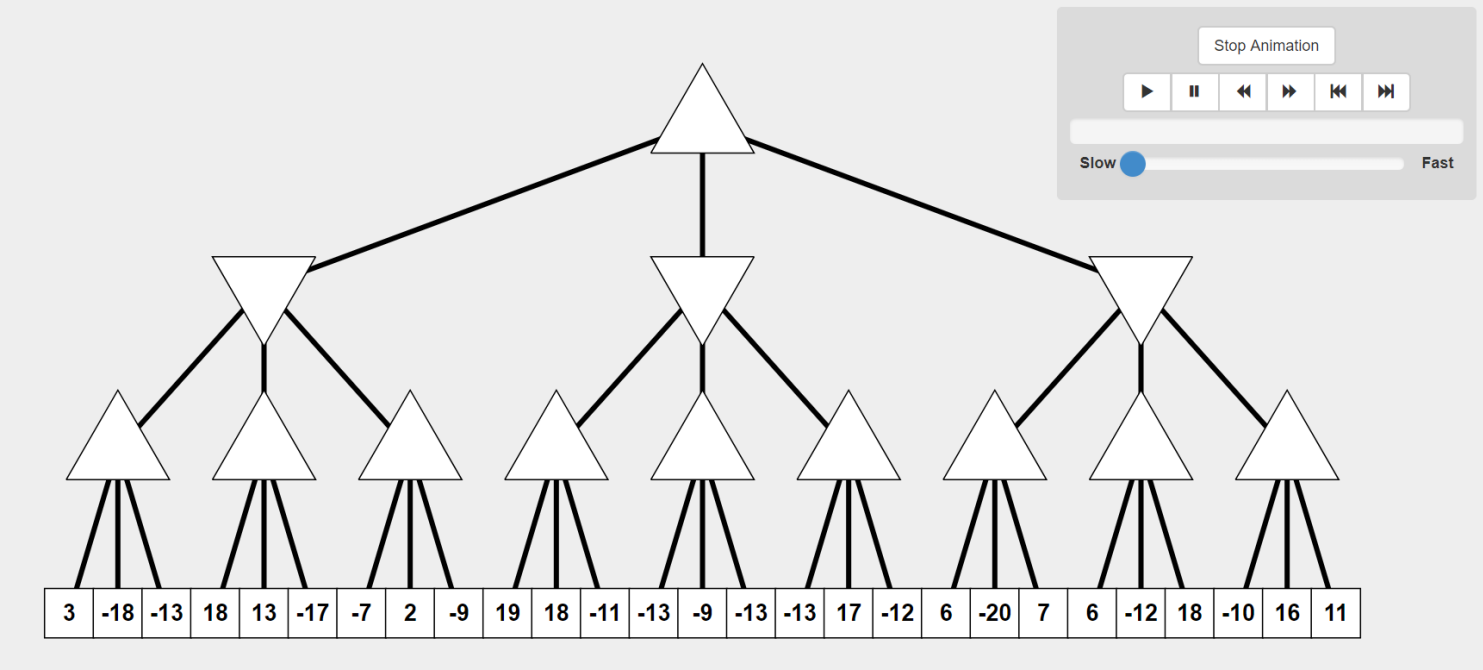
```
function MIN-VALUE(state) returns a utility value
    if TERMINAL-TEST(state) then return UTILITY(state)
    v ← ∞
    for each a in ACTIONS(state) do
        v ← MIN(v, MAX-VALUE(RESULT(s, a)))
    return v
```



Alpha-Beta Pruning

Here we consider the tree from*: *http://inst.eecs.berkeley.edu/~cs61b/fa14/ta-materials/apps/ab_tree_practice/*

We will compute the utility of each node in this tree (using **minimax**)



Alpha-Beta Pruning

Now, starting with a blank tree, like the one shown, utilize alpha-beta pruning and show:

- *The [alpha, beta] values at each node*
- *Which nodes were not expanded*