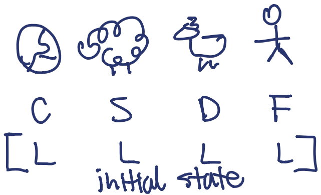
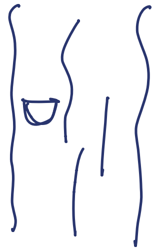
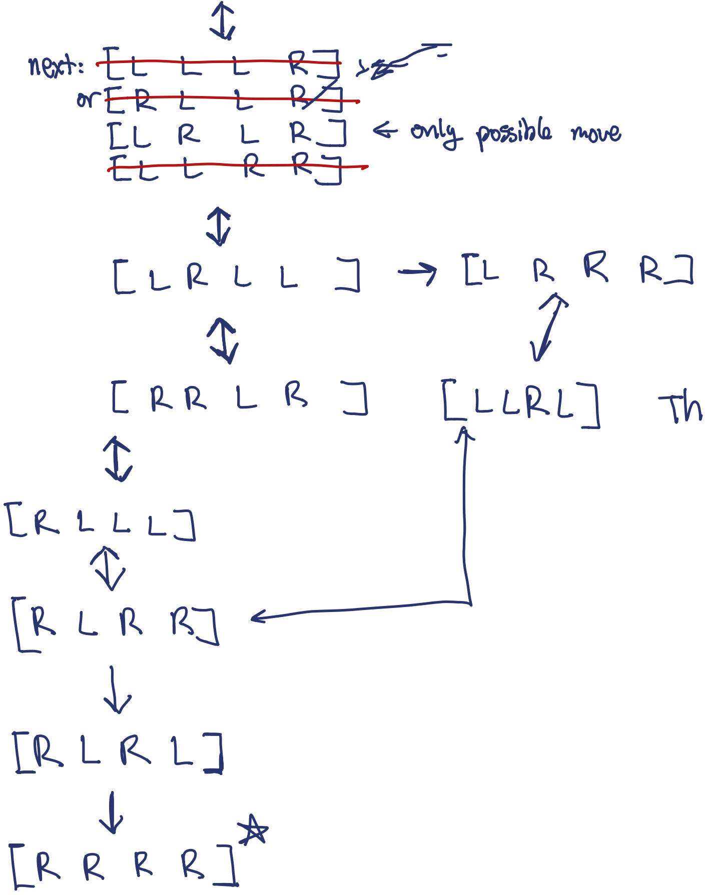
BIG idea in AI: State space

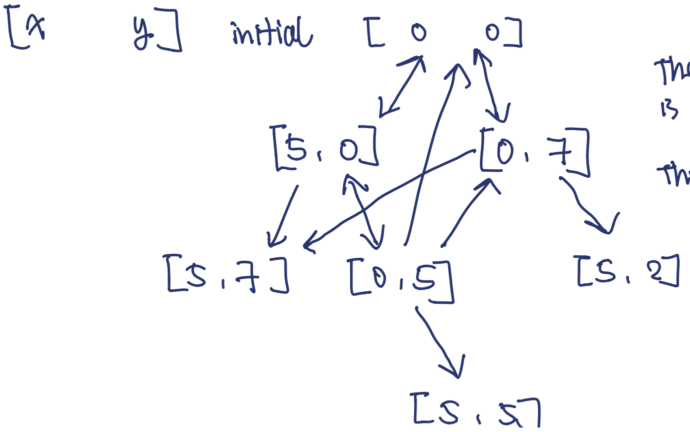






This is an undirected graph

goal state: (1, 0)



The nature of edge direction is problem-dependent

This is directed graph

Cost of Solution is important.

Start State $S_0 \in S$ state space
Goal States $G \subseteq S$
Succ(s) = set of next states
Cost(path) $= \sum_{\text{edge} \in \text{path}} \text{Cost(edge)}$

Input to our search problem

Search Algo:

```
1. $PQ \leftarrow (S_0, 0), \emptyset)$
2. Loop:
3.     $(S, C, \text{parent}) \leftarrow PQ$ [If PQ empty, fail]
4.     if $s \in G$ then success! end
5.     $PQ \leftarrow \{ t \in \text{succ}(s) : (t, c + \text{edge.cost}(s \to t)) \}, s \}$
           ↓ push all successors into the PQ.
```

① Keep backpointers
② keep a "Closed List"
(s, c, pa)