Single Agent Se

Lecture 1
Intro to State Spaces & Algori

UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

Missionaries and Cannibals

Problem due to Saul Amarel
Also known as jealous husbands

- Three missionaries and three cannibals must cross a river using a boat which can carry at most two people.
  - For both banks, if there are missionaries present on the bank, they cannot be outnumbered by cannibals (if they were, the cannibals would eat the missionaries).
  - The boat cannot cross the river by itself with no people on board.
- How do you get everyone across the river?

Solution

Left Bank	Right Bank
3M 3C	0M 0C
3M 1C	0M 2C
3M 2C	0M 1C
3M 0C	0M 3C
3M 1C	0M 2C
1M 1C	2M 2C
2M 2C	1M 1C
0M 2C	3M 1C
0M 3C	3M 0C
0M 1C	3M 2C
0M 2C	3M 1C
0M 0C	3M 3C

UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

Domain Analysis

- How many states in the state space?
- How many states are illegal?
- Problem representation?

Search Problem

- Input:
  - State space representation
  - Start state
  - Goal state (may be implicit)
- Output
  - (Optimal) path between start and g

State Space Representation

- State space is defined by either:
  - Successor function
  - Operator(s)
- Successor function
  - Given input state provides list of legal successors
- Operators
  - Given input state provides list of legal operators

State Space Representation

- Abstractly, state space is a graph G = (V, E)
  - Nodes/vertices are states
  - Edge for each successor
  - Find path in graph from start to goal
- All search algorithms are pathfinding in a graph
  - Graph might not fit in memory
- Search space can be *explicit* or *implicit*
- Goal state can be *explicit* or *implicit*

Path

- Set of states:
  - $s_1, s_2, \ldots s_i, s_{i+1}, \ldots s_N$
  - Where $s_{i+1} \in$successors(s_i)
- A *solution* if s_1 = start and s_N = goal
- *Optimal* path is the shortest path between two states

Useful paradigm

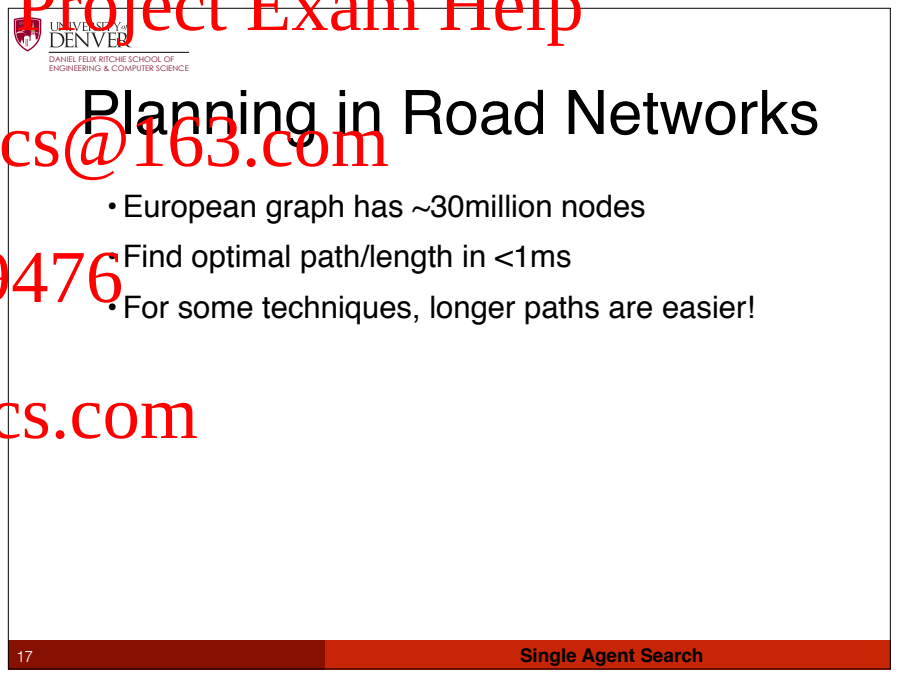
- Often the state of a program is not easily encapsulated and/or testable
- If we can:
  - query the state for possible actio
  - apply and undo actions
- It is far easier to verify the correctn implementation

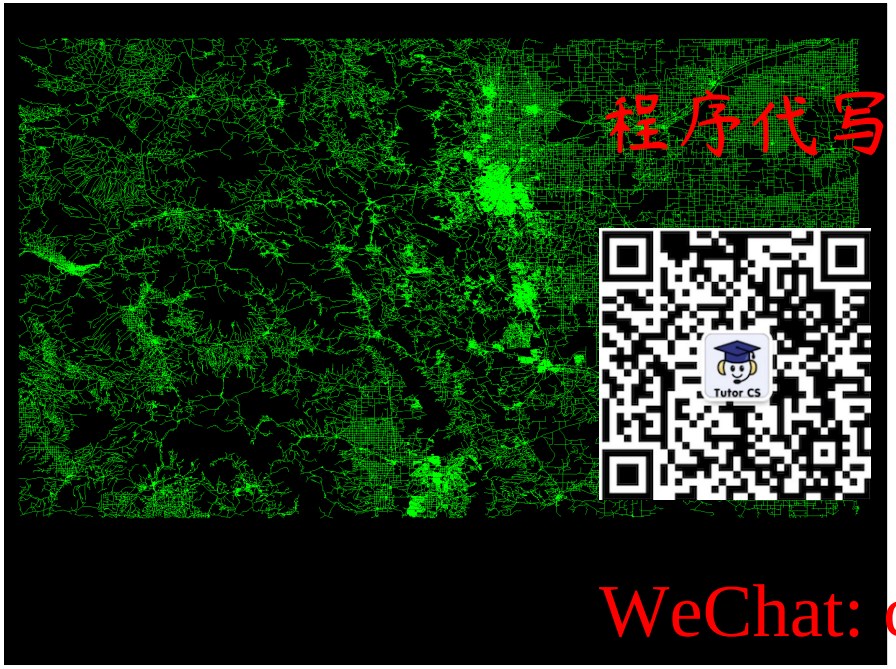
Domain Classification

- Space implicit or explicit?
- Goal implicit or explicit?
- State?
- Operator?
- Cost function?

Pathfinding

- Pathfinding
  - Commonly used in robotics, computer games, map applications
  - Find path on 2D map between start and goal
- Potentially have multiple units and/or cooperation
- Potential have additional dimensions
  - Speed, heading

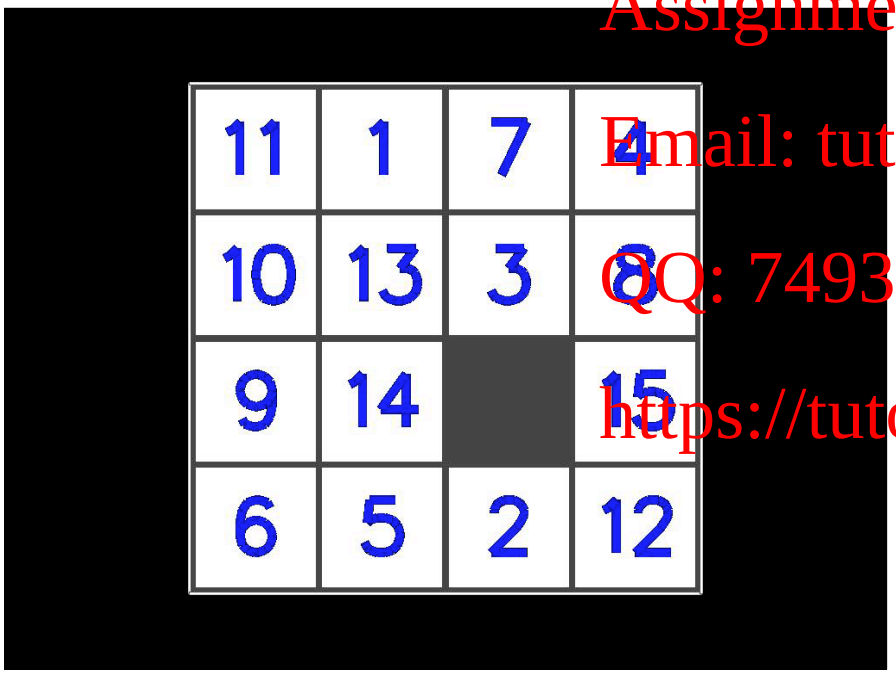




UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

Sliding Tile Puzzle

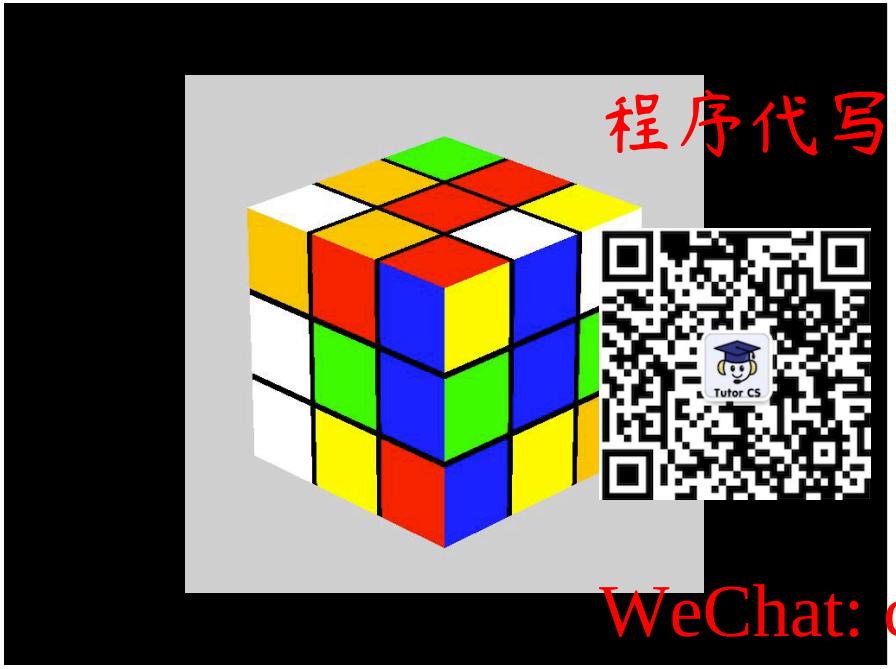
- Simple to represent, difficult to solve
- 8-puzzle has 9! = 362,880 states
- 15-puzzle has 16! = 10^{13} states
  - Half unreachable from the start
- Sam Loyd claimed to invent in the 1870's (*false*)
  - $1000 cash prize for solving

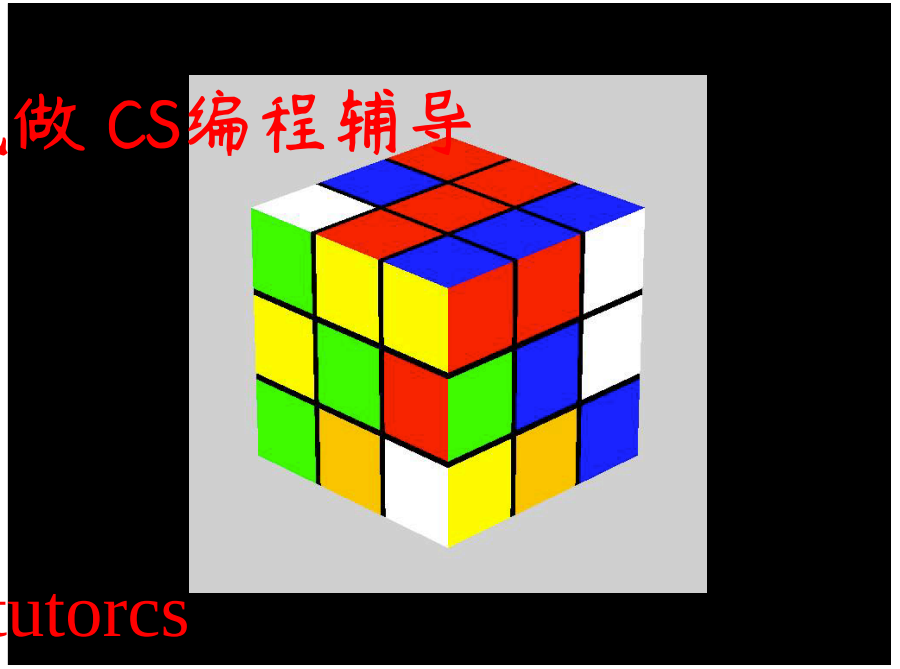


UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

Rubik's Cube

- Invented in 1974 by Ernő Rubik
- 4.3×10^{19} states
- First solved optimally in 1997 (Korf)
- Up to 17 CPU-days to solve one instance
- 100 CPU-years if using previous techniques
- ≤ 20 moves to solve **any** position





UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

Rush Hour

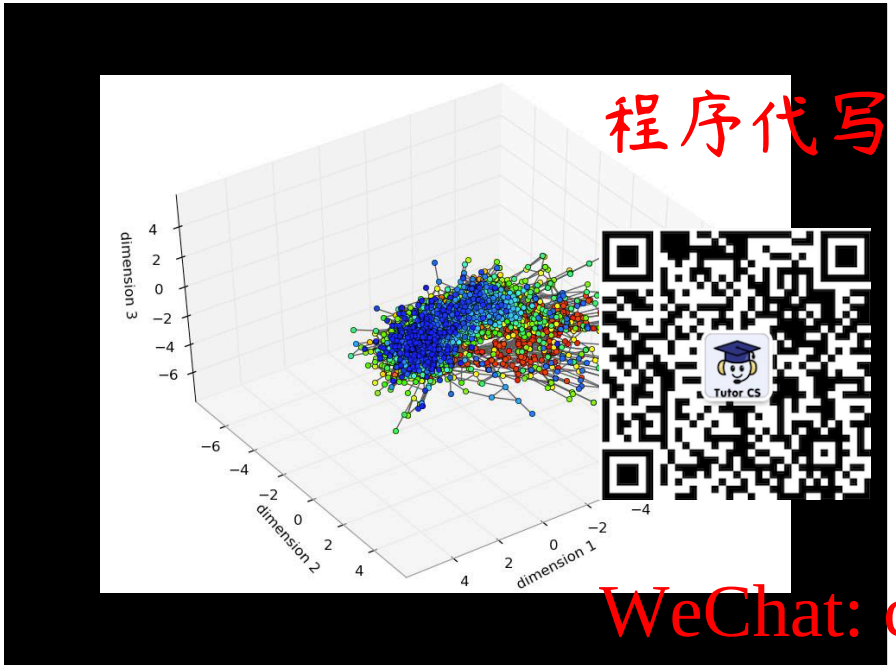
- Get red car out of maze
- Relatively easy for computer to solve

UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

Edit Distance

- FOUR
- FIVE

FOUR -> FOUL -> FOIL -> FAIL -> FALL -> FILL -> FILE -> FIVE



UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

Multiple Sequence Alignment

- A set of N sequences of DNA/RNA/protein
- A cost for matching, mismatching, blank
- Find optimal path through n-dimensional hyper-cube

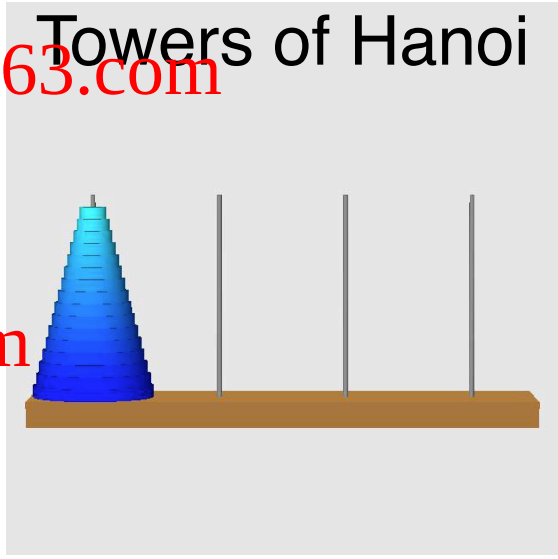
```
5'-GAATGTCCTTTCTCTAAGTCCTAAG
3'-GGAGACTTACAGGAAAGAGATTCAGGATTCAGGAGGCCTACCATGAAGATCAAG-5'

5'-GAATGTCCTTTCTCTAAGTCCTAAGTCCTCCG
3'-GGAGACTTACAGGAAAGAGATTCAGGATTCAGGAGGCCTACCATGAAGATCAAG-5'

5'-GAATGTCCTTTCTCTAAGTCCTAAGTCCTCCGG
3'-GGAGACTTACAGGAAAGAGATTCAGGATTCAGGAGGCCTACCATGAAGATCAAG-5'

5'-GAATGTCCTTTCTCTAAGTCCTAAGTCCTCCGGATG
3'-GGAGACTTACAGGAAAGAGATTCAGGATTCAGGAGGCCTACCATGAAGATCAAG-5'

5'-GAATGTCCTTTCTCTAAGTCCTAAGTCCTCCGGATGG
3'-GGAGACTTACAGGAAAGAGATTCAGGATTCAGGAGGCCTACCATGAAGATCAAG-5'

5'-GAATGTCCTTTCTCTAAGTCCTAAGTCCTCCGGATGGTACTTCTAG
3'-GGAGACTTACAGGAAAGAGATTCAGGATTCAGGAGGCCTACCATGAAGATCAAG-5'
```


UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

Top Spin

- Get tiles in order

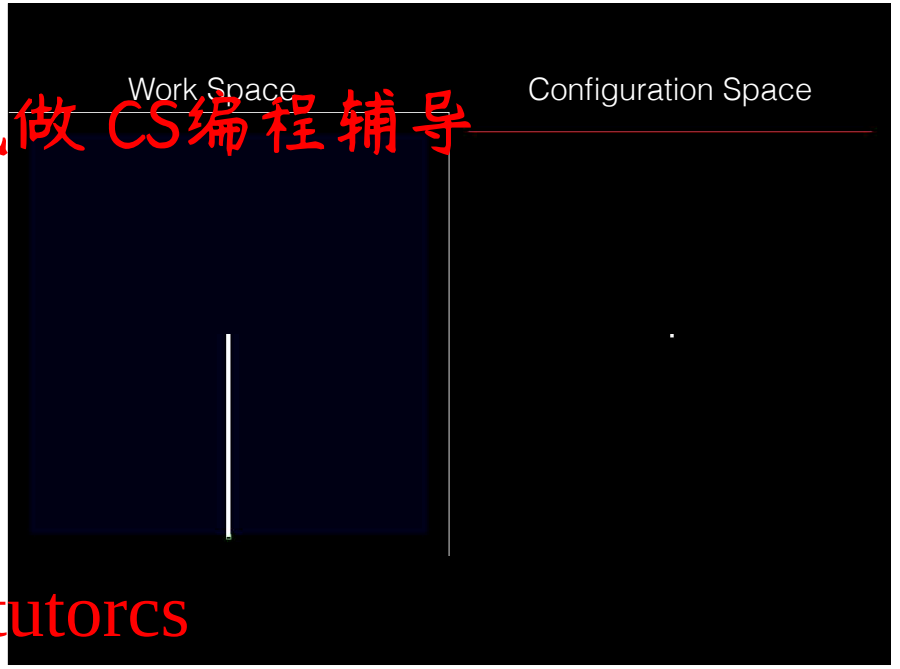
Towers of Hanoi



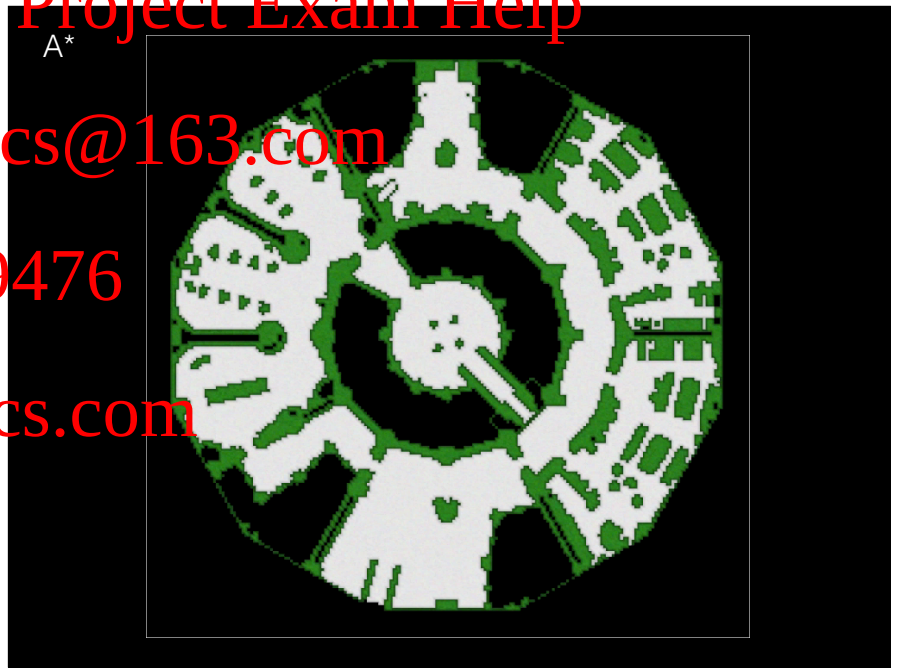
UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

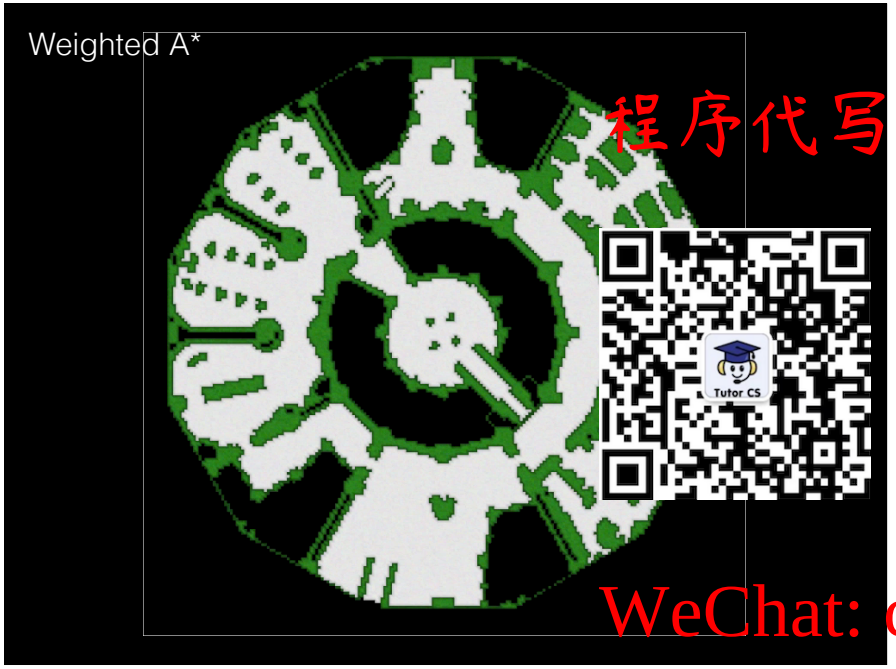
Robotic Arm

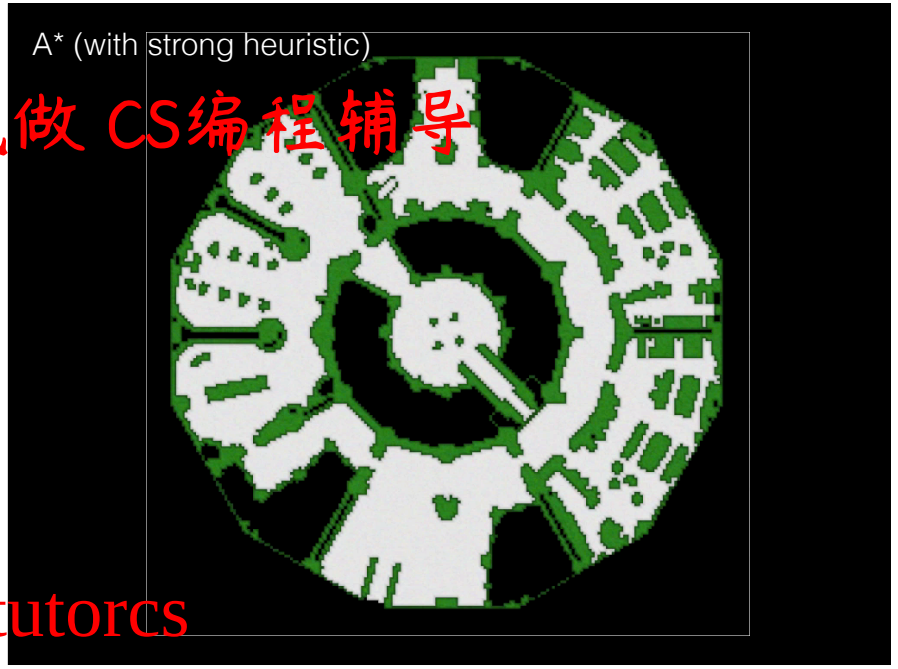
- Move the tip of the arm to a desired location
- State is location of the arms



Sample Algorithms







UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

CS (Graph Theory) vs. AI

- Graph theory also looks at **explicit** graphs
  - |V| vertices
  - |E| edges (At most $O(|V|^2)$ edges)
- AI often uses **implicit** graphs
  - Duplicate nodes hard to detect
- AI often assumes a constant branching factor

UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

CS (Graph Theory) vs. AI

- CS considers time polynomial in (nodes + edges) efficient
- AI (usually) considers this too expensive
  - representation of implicit states takes log(nodes+edges)

UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

CS (Graph Theory) vs. AI

- CS is willing to have everything stored in memory
  - polynomial in (nodes+edges)
- AI considers this exponential memo