

# Introduction to Artificial Intelligence

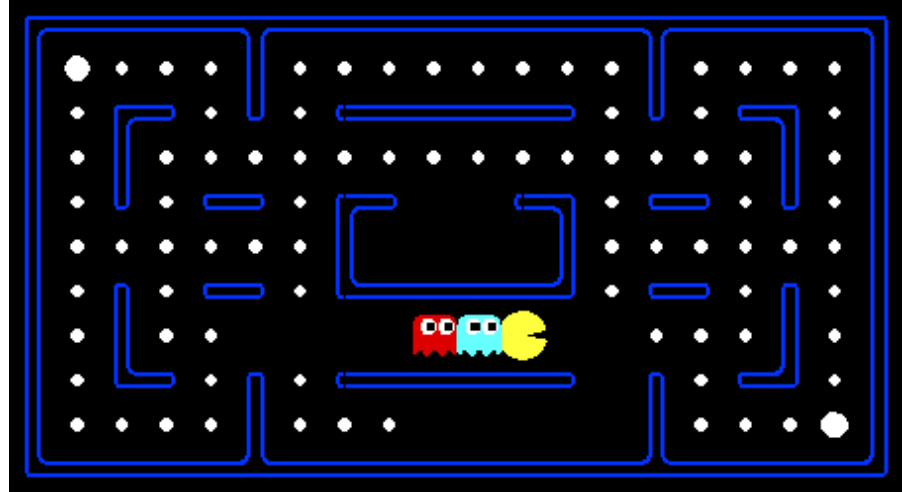
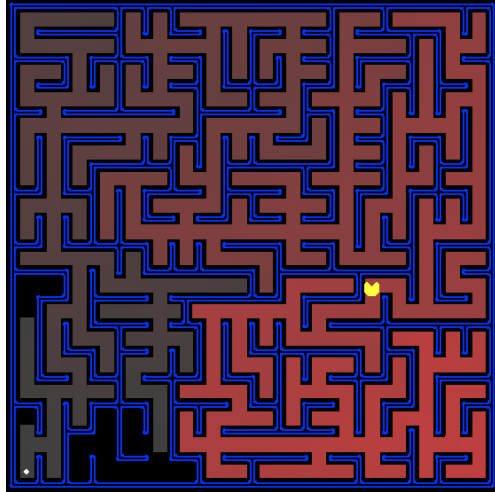
## Project 1 - Search

Jianmin Li

Department of Computer Science and Technology  
Tsinghua University

Spring, 2024

# Search



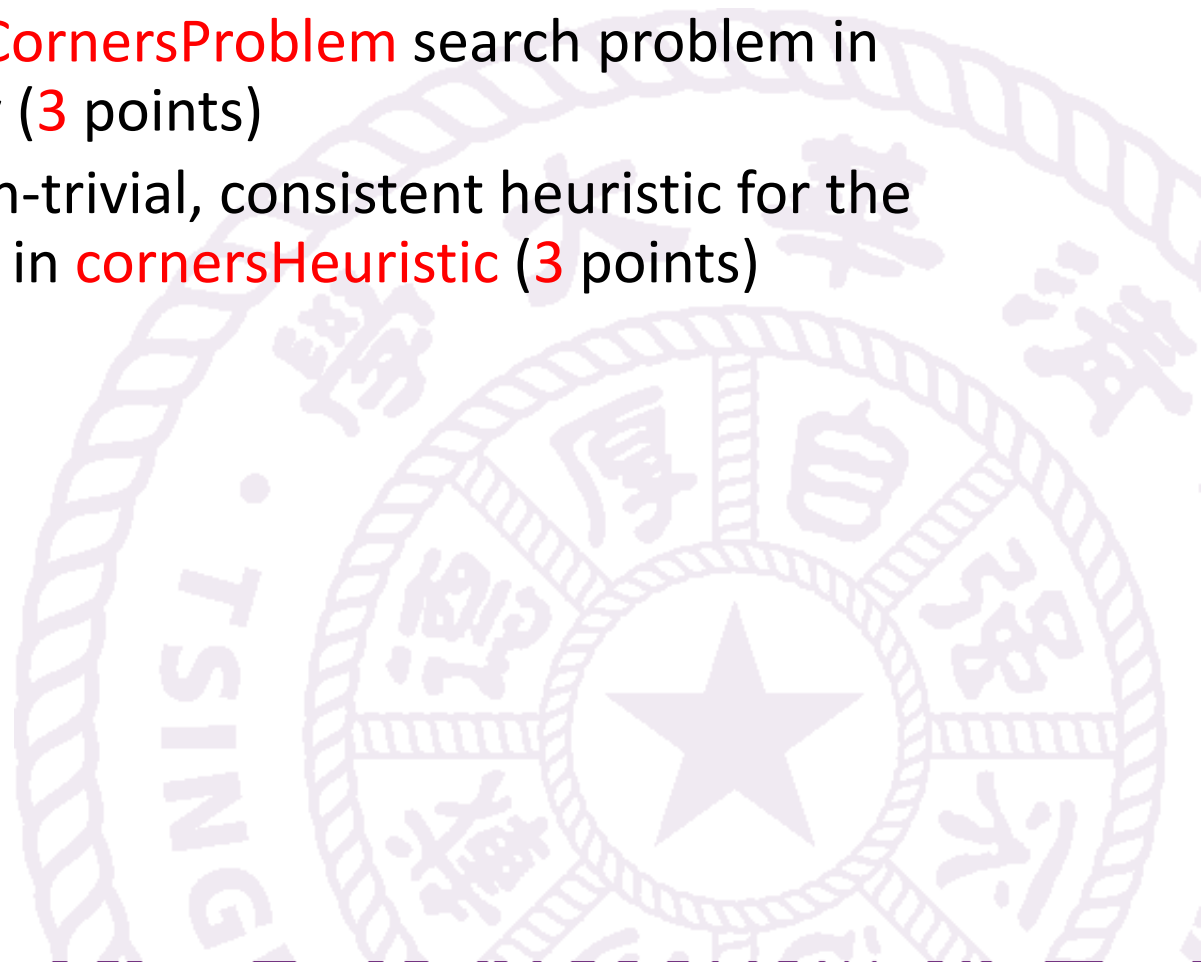
- Berkeley Pac-Man Project 1&2
  - <https://inst.eecs.berkeley.edu/~cs188/sp20/project1/>
  - <https://inst.eecs.berkeley.edu/~cs188/sp20/assets/files/search.zip>
  - <https://inst.eecs.berkeley.edu/~cs188/sp20/project2/>
  - <https://inst.eecs.berkeley.edu/~cs188/sp20/assets/files/multiagent.zip>

# Basic Tasks -- search

- Finding a Fixed Food Dot
  - Implement DFS algorithm in the **depthFirstSearch** function in search.py (2 points)
  - Implement BFS algorithm in the **breadthFirstSearch** function in search.py (2 points)
  - Implement the uniform-cost graph search algorithm in the **uniformCostSearch** function in search.py (2 points)
  - Implement A\* graph search in the empty function **aStarSearch** in search.py (3 points)

# Basic Tasks -- search

- Finding All the Corners
  - Implement the **CornersProblem** search problem in searchAgents.py (3 points)
  - Implement a non-trivial, consistent heuristic for the CornersProblem in **cornersHeuristic** (3 points)



# Basic Tasks -- multi-agent search

- MinimaxAgent
  - Implement **minimax** algorithm for any number of ghosts in the provided MinimaxAgent class stub in multiAgents.py (5 points)
- AlphaBetaAgent
  - Implement **alpha-beta** pruning algorithm in the provided AlphaBetaAgent (5 points)

# Bonus

- Eating All The Dots -- search
  - Fill in **foodHeuristic** in searchAgents.py with a consistent heuristic for the FoodSearchProblem (2 points)
- ExpectimaxAgent -- multi-agent search
  - Implement the **ExpectimaxAgent** (2 points)

# Submission

- A 3-4 pages report (either Chinese or English)
  - Compare how these algorithms perform in Pac-Man environment, e.g. state numbers, time, etc
  - Discussion
- Zip the files as the following structure
  - student\_id.zip (e.g. 20090112xx.zip)
    - student\_id.pdf
    - search.py
    - searchAgents.py
    - multiAgents.py

# Grading

- Due
  - 2024/5/1 23:59:59
- Correctness of algorithms (80%)
  - Different layouts
  - Check whether your search algorithm returns the right action sequence
- Report (20%)
- Policy
  - Discussion is encouraged , but must be written up individually
  - Do not copy/lend solution from/to others



谢谢！

