CS 188 Discussion 1:

Uninformed Search and State Spaces

Kenny Wang (kwkw@berkeley.edu)
Wed Aug 30, 2023

Who am I?

- Junior, EECS major
- Bay Area native
- can impersonate Peyrin
- not a morning person (zzzzzz)
- occasionally plays badminton
- pretty good at whistling
- Mark Rober’s selfie friend

Hate icebreakers? Sorry …

- What’s your name?
- What’s your grade?
- Is cereal a soup?

Administrivia

- Homework 0 (optional!) and Project 0 (optional!) are “due” today
- Need extensions? We will give you extensions!
- Discussion extra credit: possibly 1-2%, more details in the future

Today's Topics

- Agents
- Search Problems (Q1)
  - State Spaces
- Uninformed Search (Q2)
  - Depth-First Search
  - Breadth-First Search
  - Uniform Cost Search

Agents

- Reflex agents
  - Choose action based on current world state without considering future consequences
  - Consider how the world is
- Planning agents
  - Choose action based on hypothesized consequences of actions
  - Consider how the world would be

Search Problems

A search problem consists of

- A state space
  - set of all possible states in world
- A successor function (with actions, costs)
  - function that takes in state and action that returns successor state and cost of performing action
- A start state
- A goal test
  - function that tells us if the goal has been reached

Examples:

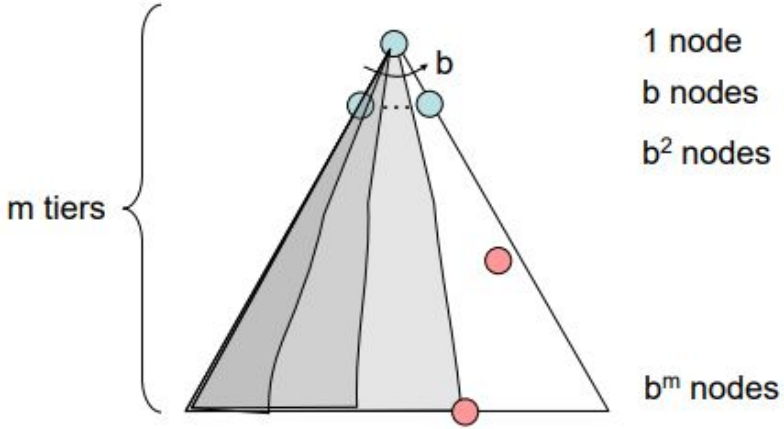
- **Pathing**
  - States: (x,y) locations
  - Actions: North, South, East, West
  - Successor: Update location only
  - Goal test: Is (x,y)=END?

- **Eat-all-dots**
  - States: (x,y) location, dot booleans
  - Actions: North, South, East, West
  - Successor: Update location and booleans
  - Goal test: Are all dot booleans false?

Worksheet Q1

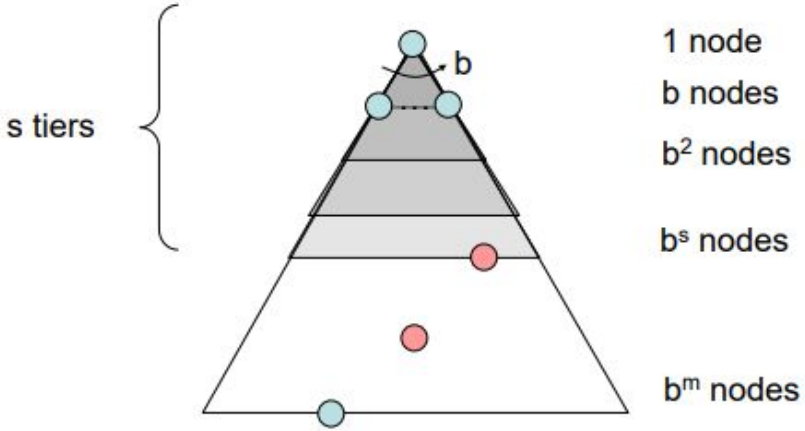
Depth-First Search (DFS)

- Strategy: Expand deepest node first
- Fringe: Stack (LIFO)
- Complete: No*
  - if graph has cycles, could keep cycling forever
- Optimal: No
  - finds the “leftmost” solution, regardless of depth or cost



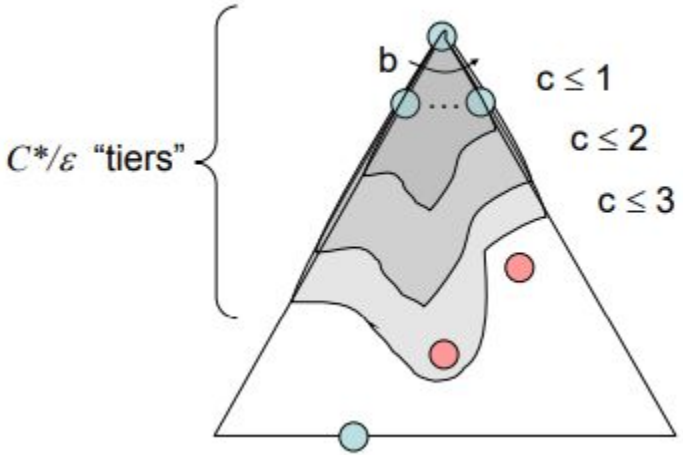
Breadth-First Search (BFS)

- Strategy: Expand shallowest node first
- Fringe: Queue (FIFO)
- Complete: Yes
  - if a solution exists, s is finite
- Optimal: Sometimes
  - only guaranteed if all costs are the same



Uniform Cost Search (UCS)

- Strategy: Expand cheapest node first
  - like Dijkstra's algorithm from CS 61B
- Fringe: Priority Queue
- Complete: Yes
  - assuming best solution has finite cost
- Optimal: Yes



Worksheet Q2

Search Algorithms (summarized)

Depth-First Search

- Strategy: Expand deepest node first
- Fringe: Stack (LIFO)
- Complete: No*
  - if graph has cycles, could keep cycling forever
- Optimal: No
  - finds the “leftmost” solution, regardless of depth or cost

Breadth-First Search

- Strategy: Expand shallowest node first
- Fringe: Queue (FIFO)
- Complete: Yes
  - if a solution exists, s is finite
- Optimal: Sometimes
  - only guaranteed if all costs are the same

Uniform Cost Search

- Strategy: Expand cheapest node first
  - like Dijkstra’s algorithm from CS 61B
- Fringe: Priority Queue
- Complete: Yes
  - assuming best solution has finite cost
- Optimal: Yes

Summary

- Agents
- Search Problems (Q1)
  - State Spaces
- Uninformed Search (Q2)
  - Depth-First Search
  - Breadth-First Search
  - Uniform Cost Search

Thank you for attending!

Attendance link:

- https://tinyurl.com/cs188fa23

My email:

- kwkw@berkeley.edu