CS343
Artificial Intelligence

Prof: Peter Stone

Department of Computer Science
The University of Texas at Austin

Good Morning, Colleagues

Good Morning, Colleagues

Are there any questions?

Logistics

- Final

Logistics

- Final
- Stephanie Tellex talk tomorrow

Logistics

- Final
- Stephanie Tellex talk tomorrow
- Multiagent assignment questions?

Search Assignment

- Question 6 – 692 nodes (10 way tie)

Search Assignment

- Question 6 – 692 nodes (10 way tie)

- Question 7 – 6866 nodes Cody Freitag;

Search Assignment

- Question 6 – 692 nodes (10 way tie)
- Question 7 – 6866 nodes Cody Freitag; 6711 nodes Robert Perce;

Search Assignment

- Question 6 – 692 nodes (10 way tie)

- Question 7 – 6866 nodes Cody Freitag; 6711 nodes Robert Perce; 6285 nodes Nicholas Moreles*

Search Assignment

- Question 6 – 692 nodes (10 way tie)
- Question 7 – 6866 nodes Cody Freitag; 6711 nodes Robert Perce; 6285 nodes Nicholas Moreles*
- Contest
  - Mark Mansi: Score - 2380

Search Assignment

- Question 6 – 692 nodes (10 way tie)
- Question 7 – 6866 nodes Cody Freitag; 6711 nodes Robert Perce; 6285 nodes Nicholas Moreles*
- Contest
  - Mark Mansi: Score - 2380
  - Matt Swarthout: Score - 2408

Search Assignment

- Question 6 – 692 nodes (10 way tie)

- Question 7 – 6866 nodes Cody Freitag; 6711 nodes Robert Perce; 6285 nodes Nicholas Moreles*

- Contest
  - Mark Mansi: Score - 2380
  - Matt Swarthout: Score - 2408
  - Tianyu Cheng: Score - 2416

Search Assignment

- Question 6 – 692 nodes (10 way tie)
- Question 7 – 6866 nodes Cody Freitag; 6711 nodes Robert Perce; 6285 nodes Nicholas Moreles*
- Contest
  - Mark Mansi: Score - 2380
  - Matt Swarthout: Score - 2408
  - Tianyu Cheng: Score - 2416
  - Liam Eagen: Score - 2418

Search Assignment

- Question 6 – 692 nodes (10 way tie)
- Question 7 – 6866 nodes Cody Freitag; 6711 nodes Robert Perce; 6285 nodes Nicholas Moreles*
- Contest
  - Mark Mansi: Score - 2380
  - Matt Swarthout: Score - 2408
  - Tianyu Cheng: Score - 2416
  - Liam Eagen: Score - 2418

Some Context

- **First weeks:** search (BFS, A*, minimax, alpha-beta)
  - Find an optimal plan (or solution)

Some Context

- **First weeks:** search (BFS, A*, minimax, alpha-beta)
  - Find an optimal plan (or solution)
  - Best thing to do from the current state
  - Assume we know transition function and cost (reward) function

Some Context

- **First weeks:** search (BFS, A*, minimax, alpha-beta)
  - Find an optimal plan (or solution)
  - Best thing to do from the current state
  - Assume we know transition function and cost (reward) function
  - Either execute complete solution (deterministic) or search again at every step

Some Context

- **First weeks:** search (BFS, A*, minimax, alpha-beta)
  - Find an optimal plan (or solution)
  - Best thing to do from the current state
  - Assume we know transition function and cost (reward) function
  - Either execute complete solution (deterministic) or search again at every step
- **This week:** MDPs —

Some Context

- **First weeks:** search (BFS, A*, minimax, alpha-beta)
  - Find an optimal plan (or solution)
  - Best thing to do from the current state
  - Assume we know transition function and cost (reward) function
  - Either execute complete solution (deterministic) or search again at every step
- **This week:** MDPs — towards reinforcement learning

Some Context

- **First weeks:** search (BFS, A*, minimax, alpha-beta)
  - Find an optimal plan (or solution)
  - Best thing to do from the current state
  - Assume we know transition function and cost (reward) function
  - Either execute complete solution (deterministic) or search again at every step
- **This week:** MDPs — towards reinforcement learning
  - Still know transition and reward function

Some Context

- **First weeks:** search (BFS, A*, minimax, alpha-beta)
  - Find an optimal plan (or solution)
  - Best thing to do from the current state
  - Assume we know transition function and cost (reward) function
  - Either execute complete solution (deterministic) or search again at every step
- **This week:** MDPs — towards reinforcement learning
  - Still know transition and reward function
  - Looking for a **policy** — optimal action from every state

Some Context

- **First weeks:** search (BFS, A*, minimax, alpha-beta)
  - Find an optimal plan (or solution)
  - Best thing to do from the current state
  - Assume we know transition function and cost (reward) function
  - Either execute complete solution (deterministic) or search again at every step
- **This week:** MDPs — towards reinforcement learning
  - Still know transition and reward function
  - Looking for a **policy** — optimal action from every state
- **Next week:** Reinforcement learning

Some Context

- **First weeks:** search (BFS, A*, minimax, alpha-beta)
  - Find an optimal plan (or solution)
  - Best thing to do from the current state
  - Assume we know transition function and cost (reward) function
  - Either execute complete solution (deterministic) or search again at every step
- **This week:** MDPs — towards reinforcement learning
  - Still know transition and reward function
  - Looking for a **policy** — optimal action from every state
- **Next week:** Reinforcement learning
  - Optimal policy without knowing transition or reward function

Pending Questions

- episodic -> continuous?
- stationary policy vs. prefs?
- optimal policy independent of start - requires discount? Or just inf H?
- What makes a policy proper?
- Who imposes finite H?
- What if |S| is too large?
- VI convergence?
- VI? PI? how choose?
- 17.10(c)
- How choose initial policy?
- Modified PI - lose optimality?