

## Adversarial Games: Entry survey

- What algorithm would be useful in games? Give examples with two different algorithms you learned in class.

Answer: (See class notes for lecture 17 for many examples).

- How would you help an agent solve a problem against an adversary? Think of a game like chess or checkers for starters.

Answer: Can use machine learning to learn adversary's moves/strategy. Can also model adversary's action as stochasticity into a tree search.

## Exit survey

- How do you reduce the tree search complexity of a turn-by-turn game like chess?

Answer: Use  $\alpha/\beta$  pruning to reduce width and use an evaluation function to cutoff the depth of search tree.

- Give an example for a game that we haven't studied in class which can be solved with the minimax algorithm. Suggest an evaluation function at the cutoff nodes.

Answer: Backgammon, Checkers, Othello: can count your pieces.  
Risk/Diplomacy: can count territories and armies. Four in a Row: whether next move is winning and count of 3-in-a-row pieces.