Game Theory

Victor Cao, Jason Zhang

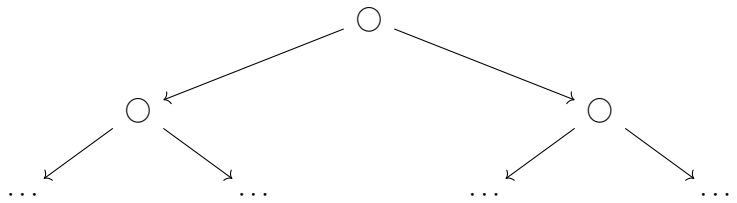
April 24, 2026

1 Introduction

The purpose of this lecture is to cover finite deterministic games that might appear in competitive programming contexts. Unfortunately, game theory in competitive programming contexts is EXTREMELY limited. Most actual game theory topics (relating to combinatorial or not) will never be seen in a competitive programming problem, or they are too rare to learn for the sake of competitive programming. The only real theory in competitive programming game theory is that of NIM and Sprague-Grundy, which will not be covered in this lecture. If you are interested, you can look at this resource: `https://cp-algorithms.com/game_theory/sprague-grundy-nim.html`

1.1 Rough Tips

Firstly, it is important to interpret games as trees. Each game state is a node on the tree, and the edges represent actions to take. It is important NOT to implement this interpretation in actualized code. Instead, one should simply think of problems this way.



Sometimes, it is useful to think of problems not explicitly stated to be games as games. You can make a connection with the above tree diagram. Here are some other things that are occasionally useful:

1. An *invariant* is some property (about either a strategy or position) that does NOT change after some move, transformation, etc.
2. Another important concept is that of symmetry, in which the game position is independent of who is playing.
3. This can give rise to strategy stealing, a case in which a player can adopt the strategy of another to win.

1.2 Greedy Strategies

One particular notable type of strategy frequently seen is that of the greedy strategy. The greedy strategist picks out the most locally optimal move, ignoring long-term consequences. This sometimes works. As always, one develops an intuition when to use greedy after solving a lot of problems.

Sometimes, one can show a greedy strategy is optimal in the same way that one might show that a greedy algorithm is optimal. In these cases, one might look to an argument similar to the *exchange* argument. However, competitive programmers generally show aversion to real theoretical computer science topics so it will not be covered in this lecture. See this `https://www.cs.cornell.edu/courses/cs482/2007su/exchange.pdf` if you are interested.

2 Practice Problems

Here are some practice problems, in no particular order.

- `https://codeforces.com/problemset/problem/2203/D`

- https://codeforces.com/contest/1404/problem/B
- https://codeforces.com/contest/1404/problem/D
- https://codeforces.com/contest/1375/problem/F
- https://codeforces.com/problemset/problem/2200/A
- https://codeforces.com/problemset/problem/2200/E
- https://codeforces.com/problemset/problem/2199/A
- https://codeforces.com/problemset/problem/2196/A
- https://codeforces.com/problemset/problem/2184/D