TCSS 343 - Week 7

Jake McKenzie

August 30, 2018

Greedy Algorithms

"Finding a needle in a haystack is actually quite easy. It's finding a very specific piece of hay in a haystack that happens to be hard."

...

Avi Wigderson

0. With this problem I want to present a problem to you and ask you why the greedy algorithm fails.

Imagine we have a wizard that knows a few spells. Each spell has 3 attributes: Damage, cooldown time, and a cast time.

Cooldown time: the amount of time (t) it takes before being able to cast that spell again. A spell goes on "cooldown" the moment it begins casting.

Cast time: the amount of time (t) it takes to use a spell. While the wizard is casting something another spell cannot be cast and it cannot be canceled.

The question is: **How would you maximize damage given different sets of spells?**

It is easy to calculate the highest damage per cast time. But what about in situations where it is better to wait then to get "stuck" casting a low damage spell when a much higher one is available...for example consider the two sets of spells:

Spell 0. 100 damage at a rate of 1 second per cast with a 10 second cooldown.

Spell 1. 10 damage at a rate of 4 second per cast with a 0 second cooldown.

Optimal spell ordering $\Sigma = \{\text{Spell 0, Spell 1, Wait, Repeat}\}$

F. Given an arbitary amount of time t what is the maximum amount of spells we can cast S?