HOBTorrent Challenge Tutor Guide

Introduction

Short Version

Introduce the students to the scenario. If you're wanting a dramatic way of doing this, see the extended version ;)

Give students the link to the tute sheet: `hobweb.github.io`. This isn't on blackboard so students later in the week don't get an advantage (ask students not to distribute this for that reason).

Once you've introduced the scenario, get students to split into groups. Groups of 2-4 is probably a good number, but you can play around with this depending on how many people are in your tute (eg. a small tute should have smaller groups). Get groups to make a "company name", but don't get them to spend too much time on it. Students will then have 40-45 minutes to work on the task. Help them out but don't give them the answers. As groups complete tasks check them off (I'd put something up on the whiteboard to record the points groups are getting). It's very unlikely any of the teams finish all tasks (I'm only expecting each average team to finish two tasks). I'd also recommend that students don't attempt the last problem unless they are feeling really brave.

Extended Version

Talk to Henry if you want to make the introduction really exciting.

Victory Points Scheme

Put a table like the following on the whiteboard and continuously update the points in it as the tute progresses.

Team	Core	Comm	Cleanup	Upload
[name]	/2	/3	/5	/5

Mark each task as follows:

Core Functionality (2 points)

- (+1 points) An attempt has been made for all three methods
- (+1 points) All methods are expected constant-time (i.e. hash sets are used)

Network Communication (3 points)

- (+1 points) The class is modified to behave as a graph, and is implemented efficiently (NOT as an adjacency matrix)
- (+1 points) Some sort of pseudocode is written to do a traversal over the graph (either BFS or DFS)
- (+1 points) BFS is used to find the shortest path and the team can explain why BFS works well

File System Cleanup (5 points)

See: `https://leetcode.com/problems/maximum-frequency-stack/`
This one is easy to get a solution, but hard to make efficient. Everything can be done in constant-time if thought about correctly.

- (+1 point) An attempt was made that appears correct (doesn't have to be efficient)
- (+1 point) A correct attempt was made, *cleanup* is constant-time (*addFile* can be worse)
- (+3 points) A correct attempt was made, *cleanup* is constant-time and *addFile* remains constant-time

File Upload (5 points)

See: `https://leetcode.com/problems/capacity-to-ship-packages-within-d-days/`
This is probably the hardest to think about.

- (+2 points) An attempt was made that appears correct
- (+3 points) An efficient attempt was made that appears correct (i.e. binary search is used)

Wrapping Up

Take the points table, and add it to the master table in the blackboard wiki. If possible, also record the names of the students in each team (so we know who the best students are ;)). I may try to organise some sort of prize to give to the winning students in week 13.