

Startup Tycoon

Algorithms with a Purpose

October 17, 2015

1 Introduction

Welcome, entrepreneurs, to the Startup Tycoon. According to your trustworthy news source on X Combinator, there has recently been a growing demand for widgets in San Franhattan. And so, like the street-smart businessperson you are, you've decided to capitalize on (and hopefully monopolize) the nascent market by strategically taking on orders and expanding Tuber, your widget business. Of course, you aren't the only one with this objective—you are joined by others who have their eyes on the prize as well. So, get out there and lead Tuber on to be great, and make Andrews Carnegie and Mellon proud!

2 Terminology

player	a single entity in a game, representing a team of up to 4 humans
game	an iteration of a set of orders and turns on a distinct map
turn	a single time step of a game during which an order may appear and a player may take an action
action	creating a station and/or responding to an order
order	a request for a widget originating from a home on the map
home	a node on the map that can generate orders for widgets
station	a node on the map that can respond to orders for widgets
map	the map on which orders appear and actions occur, consisting of a collection of nodes, some of which are connected

3 Rules

1. Every node can be reached from another through some path through the map. Nodes are connected with neighbors by **undirected** edges.
2. Each station will be located at a **player-determined** location. The player may choose when to build their stations (including the first station), provided that they have enough capital.
3. Each turn will be capped at half a second. Each game will end after **400** turns (a little over three minutes at most).

4. Orders will appear at certain time steps, with probabilities following a **Poisson distribution**. An order contains the following:
 - A destination node
 - A bounty for the request
5. You can respond to the request whenever you choose after receiving it (including the time step it was received) by returning a **set of edges** from the station to the destination. Of course, you may not respond to orders that have timed out. The actual bounty received is a function of how long it takes for the widget to reach its destination (see Section 4: Scoring).
6. For the duration of time steps equal to the length of the path returned, those edges **cannot be used** for other requests. (Imagine your widget traveling through one edge per time step, during which the entire path to be occupied while it's in transit.)
7. A **new station** can be built on any turn instantaneously and can begin serving on the same turn it is created as long as the order of the commands is correct. Stations can be built on any node in the map including homes. The cost of the station increases per number existing stations.

4 Scoring

Score per game:

$$\sum_{n \in N} s * dt - \sum_{b \in B} c$$

s = initial bounty

d = decay factor

t = time steps elapsed

N = number of fulfilled orders

c = cost of a station

B = number of stations built

Order score function is the number of orders fulfilled minus the cost of existing buildings. Note that unfulfilled orders will decay in value over time and then time out eventually, with a value of 0.

Order values decay linearly over time, and scores will reflect time taken to deliver each fulfilled order.

5 Tournament

- The competition will take place in multiple rounds where each player's score is taken individually (i.e. there will not be multiple players competing on the same map).

- A single map layout will be used for all players for a given round, and maps will change per round.
- The top 50% of players will advance at the end of each round.
- Scores will reset to 0 at the start of a round.
- The top three players are the top three players from the final match.

6 Tips

- The practice maps will not be the same as the tournament maps, but they will always be connected and undirected.
- As each map is a connected, undirected graph, graph algorithms (search and shortest-path in particular) will probably come in handy.
- You may also find it useful to look into and take ideas from various **streaming algorithms**. Wikipedia is a good place to start: (https://en.wikipedia.org/wiki/Streaming_algorithm)