CS 161 Fall 18 Section 5

September 2018

Warm-up: Greedy or Not?

Sometimes it can be tricky to tell when a greedy algorithm applies. For each problem, say whether or not the greedy solution would work for the problem. If it wouldn't work, give a counter example.

1. You have unlimited objects of different sizes, and you want to completely fill a box with as few objects as possible. (Greedy: Keep putting the largest object possible in for the space you have left)
2. You have unlimited objects, all of which are size 3^k for some integer k, and you want to completely fill a box with as few objects as possible. (Greedy: same approach as the previous problem)
3. You have lines that can fit a fixed number of characters. You want to print out a series of words while using as few lines as possible. (Greedy: Fit as many words as you can on a given line)
4. From Midterm 2: You want to get from hotel 1 to hotel n, and you can travel at most k distance between hotels before collapsing from exhaustion. Find the minimum cost of hotels. (Greedy: Go as far as you can before stopping at a hotel)

Pareto Optimal

Given a set of 2d points P, a Pareto optimal point is a point (x, y) such that $\forall(x', y')$ we have either $x > x'$ or $y > y'$. Develop an algorithm to to find all Pareto optimal points.

Cutting Ropes

Suppose we are given n ropes of different lengths, and we want to tie these ropes into a single rope. The cost to connect two ropes is equal to sum of their lengths. We want to connect all the ropes with the minimum cost.

For example, suppose we have 4 ropes of lengths 7, 3, 5, and 1. One (not optimal!) solution would be to combine the 7 and 3 rope for a rope of size 10, then combine this new size 10 rope with the size 5 rope for a rope of size 15, then combine the rope of size 15 with the rope of size 1 for a final rope of size 16. The total cost would be 10 + 15 + 16 = 41. (Note: the optimal cost for this problem is 29. How might you combine the ropes for that cost?)

Find a greedy algorithm for the minimum cost and prove the correctness of your algorithm.

Mice to Holes

There are n mice and n holes along a line. Each hole can accommodate only 1 mouse. A mouse can stay at his position, move one step right from x to $x+1$, or move one step left from x to $x-1$. Any of these moves consumes 1 minute. Mice can move simultaneously. Assign mice to holes such that the time it takes for the last mouse to get to a hole is minimized, and return the amount of time it takes for that last mouse to get to its hole.

Example:

Mice positions: 4 -4 2

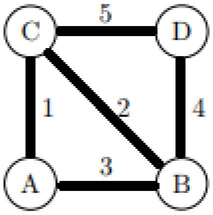
Hole positions: 4 0 5

Best case: the last mouse gets to its hole in 4 minutes $\{4 \rightarrow 4, -4 \rightarrow 0, 2 \rightarrow 5\}$ and $\{4 \rightarrow 5, -4 \rightarrow 0, 2 \rightarrow 4\}$ are both possible solutions

Minimum Spanning Trees

A minimum spanning tree (MST) or minimum weight spanning tree is a subset of the edges of a connected, edge-weighted (un)directed graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight.

Consider the below shown undirected graph with weights shown on the edges.



1. Find the MST for this graph

2. Think about how you went about finding this MST and try to form a general greedy rule based on it. (Take Home)