CS157   Lecture #8 : Greedy Algorithms

WARMUP — What is the best way to arrange positive numbers to minimize $1 \cdot 1^{st} + 2 \cdot 2^{nd} + 3 \cdot 3^{rd} \cdots$?

Ans : decreasing order

Pf : Suppose for contradiction that the set of numbers $v_1, \ldots, v_n$ are optimally arranged $\vee$ and not in decreasing order. Then there must be some v_i, v_{i+1} s.t. $v_i < v_{i+1}$. Swapping these two yields a sum of $i v_{i+1} + (i+1) v_i < i v_i + (i+1) v_{i+1}$, which means that our original arrangement was not minimal, which is a contradiction.

(can also be proved by induction)

SCHEDULING PROBLEM : Choose the schedule that allows you to go to the most events
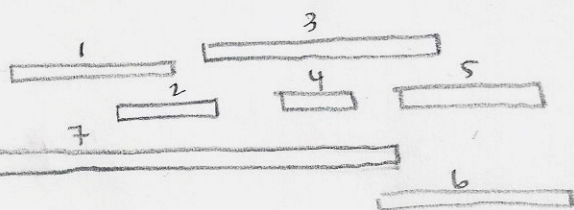
OPTIMAL : 2 4 5
         1 4 5

IDEA : Choose the class that ends soonest (1 4 5)

(or the class that starts latest → 2 4 5)

HOW DO WE PROVE THIS?

INDUCTION : There is an optimal solution where the first i choices are made like we made them

EXTREMAL : Suppose our solution not optimal. Swapping our ith choice in for the ith choice of the OPT soln is at least as good.

*How long does this take? Sort by start time + loop through events $= O(n \log n)$

* You could also solve this using **dynamic programming** – it will take more time ($\approx O(n^2)$), but is more general, in that you can use the same solution to optimize time spent

MINIMUM SPANNING TREE
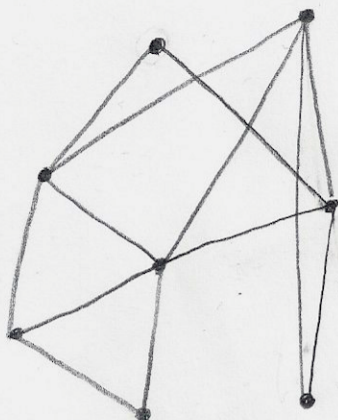
INPUT : connected graph G

OUTPUT : the shortest path that connects all vertices (no loops)

IDEA : at each step, pick the shortest edge that does not create a cycle.

PROOF : Inductive hypothesis = there is an optimal solution where the first i choices are made like we made them.

Consider the first edge we choose such that it is not in any optimal solution. Insert this edge into an optimal solution that matches our 1st $i-1$ choices. This must create a cycle. The longest edge must be part of the optimal soln since our algo considers edges in increasing length

→ removing this longest edge creates a better solution than the optimal soln, which is a contradiction ∎