CSCE 411
Shortest Paths and MSTs

---

1. Suppose that we are given a graph G and want to compute a *maximum* spanning tree–i.e. a tree which maximizes $\sum_{e \in E} w(e)$.

   (a) Suppose that we modify the Prim-Jarnik algorithm so that at each step it adds the *heaviest* edge connecting its current subgraph to the rest of G. Does this algorithm solve the maximum spanning tree problem? Justify your answer.

   (b) Suppose we modify Kruskal's algorithm so that it sorts the edges from heaviest to lightest and then repeatedly adds the first edge which does not create a cycle. Does this algorithm solve the maximum spanning tree problem?

2. The hit new role playing game Adventures in Algorithmica features n different zones $z_1, \ldots, z_n$. Each zone z_i has a minimum required level to enter, ℓ_i, which is an integer with $1 \leq \ell_i \leq n$. Fortunately, gaining experience in this game is very easy. You gain 1 level every time you enter a new zone. Unfortunately, zones take a long time to load. Curiously, the time it takes a zone to load depends on which other zone you are entering it from. You are given an array A of loading times, where $A[i, j]$ is the time that zone j takes to load if you enter it from zone i. You are writing a guide for the game and want to figure out the fastest routes to all of the high level zones starting from level 1 zones.

   Give an $O(n^4)$ algorithm which takes as input, the list of zones, list of level requirements, an array of loading times, and outputs a 2D array T where $T[i, j]$ is the minimum time to get to z_j, assuming that you start in z_i at level ℓ_i.

3. Consider the following algorithm:

```
EdgeRemover(G)
  A ← an array of G's edges, sorted in order of decreasing weight
  T ← G
  For i from 1 to m:
      If removing A[i] from T would not disconnect T:
          Remove A[i] from T
  Return T
```

   Prove that for every connected graph G with distinct edge weights, EdgeRemover(G) returns a minimum spanning tree for G.

4. Suppose that we are given a list of n variables $x_1, \ldots, x_n$ and a list of m constraints of the form: $x_i - x_j \leq c_k$, where $c_1, \ldots, c_m$ are integer constants Give an $O(mn)$ time algorithm to determine whether or not there exist integer values for the variables which satisfy all of the constraints.