CSE 332: Data Structures and Parallelism

Section 10: P/NP, Final Review

Snow Day

After 4 snow days last year, UW has decided to improve its snow response plan. Instead of doing "late start" days, they want an "extended passing period" plan. The goal is to clear enough sidewalks that everyone can get from every classroom to every other **eventually** but not necessarily very quickly.

Unfortunately, UW has access to only one snowplow. Your goal is to determine which sidewalks to plow and whether it can be done in time for the first 8:30 AM lectures.

You have a map of campus, with each sidewalk labeled with the time it will take to plow to clear it.

a) What will the vertices of your graph be?

Have a vertex for each building.

b) What will the edges be? You should at least say whether your edges are directed or not and whether they're weighted or not.

Have an edge for each section of sidewalk. The edges should be undirected, and weighted by the time it will take the snowplow to clear it.

c) What algorithm will you run on your graph?

Run an MST algorithm (either **Kruskal's** or Prim's).

d) How will you interpret the output of your algorithm? (i.e. which sidewalks to plow “in the real world" instead of just in graph terms).

Whatever edges are chosen are the sidewalks the plow should clear.

e) Briefly (2-4 sentences) explain why your model works. You should at least address why you ran the algorithm you did (e.g., why are you looking for a shortest path/MST/topological ordering/etc.) and how you are ensuring your algorithm will be able to produce an "extended passing period" plowing plan.

We want an MST because our goal is to connect everything cheaply (not find the shortest route from A to B). Look at the weight of the MST. That's how long it will take to plow. If the plow can start in time to finish by 8:30, then we can start on time!

P, NP, NP-Complete

a) "NP" stands for:

Nondeterministic Polynomial

b) What does it mean for a problem to be in NP?

Given a candidate solution to a decision problem, we can verify whether the solution is correct in polynomial time.

c) For the following problems, circle ALL the sets they (most likely) belong to:

Is there a path of weight at most *k* from one vertex to another vertex in a weighted directed graph?

[✓ NP] [✓ P] NP-complete None of these

Is there a cycle that visits each edge in a graph exactly once?

[✓ NP] [✓ P] NP-complete None of these

Will this program run forever?

NP P NP-complete [✓ None of these]

Can we find the prefix sum of an array in parallel using 10 processors?

[✓ NP] [✓ P] NP-complete None of these

Is there a path that starts and ends at the same vertex that visits every vertex exactly once?

[✓ NP] P [✓ NP-complete] None of these