Tutorial 7: Graph algorithms

1 DAGs

We saw in class that we can linearize a directed acyclic graph (DAG) using DFS and sorting the vertices by the finish/post-visit times. Recall that a vertex in a directed graph is a *source vertex* if it has in-degree 0. Here's another proposed algorithm for linearizing a DAG G:

Algorithm 1: Linearizer(G)

while G *is not empty* **do**
  Find a source vertex v in G;
  Output v;
  Remove v from G;

Does this algorithm always produce a valid linearization of DAGs? Give a proof or counter-example to justify your answer.

2 Minimum Spanning Trees

Here's another proposed greedy algorithm for computing the Minimum Spanning Tree of a weighted graph:

Algorithm 2: CycleCut(G)

Sort the edges E by decreasing weight;
for *each* $e \in E$ *in order of decreasing weight* **do**
  if e *is part of a cycle in* (V, E) **then**
    $E \leftarrow E \setminus \{e\}$;
Return $T = (V, E)$;

Does this algorithm always return a MST of weighted graphs? Give a proof of correctness or a counter-example to justify your answer.

3 Single-source shortest path

We saw a greedy algorithm (Dijkstra's algorithm) for solving the SSSP problem on graphs with non-negative weights, and a dynamic programming algorithm (Bellman-Ford algorithm) for solving the SSSP problem on weighted graphs that can have both positive and negative edge weights.

Prove that the following proposed reduction from the general SSSP problem to the SSSP problem on graphs with nonnegative weights does not correctly solve the problem:

1. Let $-M$ be the minimum edge weight in G.
2. Define G' to be the weighted graph with the same edges as G and with edge weights $w'(u,v) = w(u,v) + M$ for each $(u,v) \in E$.
3. Solve the SSSP problem for the nonnegative weighted graph G'.
4. For each vertex v with shortest path weight W in G' obtained by following a path of ℓ edges from s to v, output the distance $W - \ell M$.