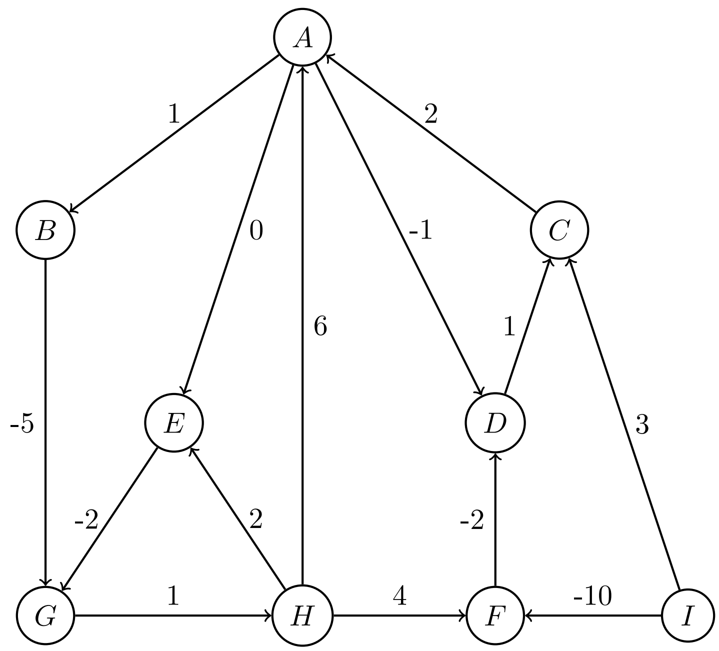
CS 170 DIS 05

Released on 2018-02-18

1 Bellman-Ford Practice

(a) Run the Bellman-Ford algorithm on the following graph, from source A. Relax edges (u, v) in lexicographic order, sorting first by u then by v.



(b) What problem occurs when we change the weight of edge (H, A) to 1? How can we detect this problem when running Bellman-Ford? Why does this work?

(c) Let $G = (V, E)$ be a directed graph. Under what condition does the Bellman-Ford algorithm returns the same shortest path tree (from source $s \in V$) regardless of the ordering on edges?

2 MST Basics

For each of the following statements, either prove or supply a counterexample. Always assume $G = (V, E)$ is undirected and connected. Do not assume the edge weights are distinct unless specifically stated.

(a) Let e be any edge of minimum weight in G. Then e must be part of some MST.

(b) If e is part of some MST of G, then it must be a lightest edge across some cut of G.

(c) If G has a cycle with a unique lightest edge e, then e must be part of every MST.

(d) For any $r > 0$, define an r-path to be a path whose edges all have weight less than r. If G contains an r-path from s to t, then every MST of G must also contain an r-path from s to t.

3 Minimum Spanning Trees (short answer)

(a) Given an undirected graph $G = (V, E)$ and a set $E' \subset E$ briefly describe how to update Kruskal's algorithm to find the minimum spanning tree that includes all edges from E'.

(b) Suppose we want to find the minimum cost set of edges that suffices to connect a given weighted graph $G = (V, E)$; if the weights are non-negative then we know that the optimum will be a MST. What about the case when the weights are allowed to be negative? Does it have to be a tree if the weights are allowed to be negative? If not, how would you find this minimum-cost connected subgraph?

(c) Describe an algorithm to find a maximum spanning tree of a given graph.

4 Picking a Favorite MST

Consider an undirected, weighted graph for which multiple MSTs are possible (we know this means the edge weights cannot be unique). You have a favorite MST, F. Are you guaranteed that F is a possible output of Kruskal's algorithm on this graph? How about Prim's? In other words, is it always possible to "force" the MST algorithms to output F without changing the weights of the given graph? Justify your answer.