6.046 Meta Review Session

Quiz 1

Fall 2014, Problem 2

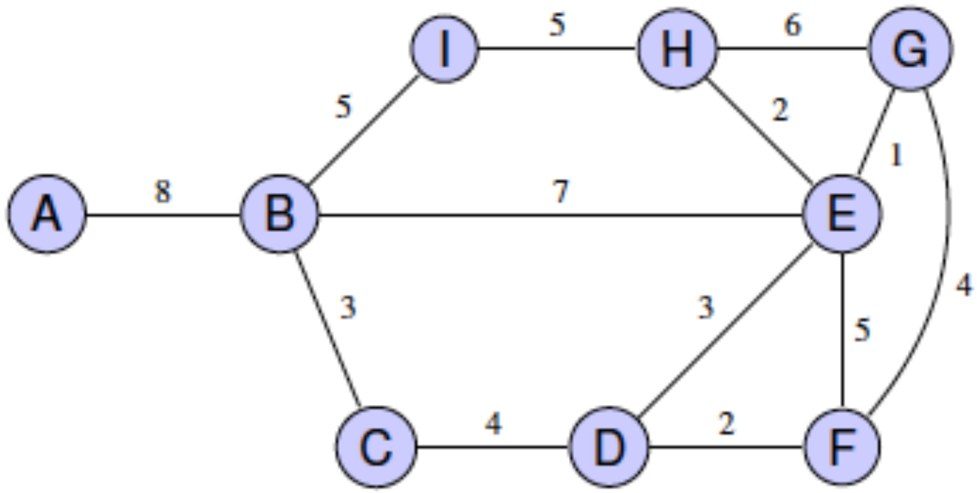


Figure 1: Example graph for Problems 2(a) and 2(b).

Problem 2: MST *[20 points]*

(a) *[3 points]* In the graph shown in Figure 1, prove that the edge between B and E is not in **any** minimum spanning tree.

(b) *[3 points]* In the graph shown in Figure 1, prove that the edge between E and G is in **every** minimum spanning tree.

Fall 2014, Problem 2

(c) *[7 points]* Suppose that we are given a minimum spanning tree T of a graph $G = (V, E)$ that is connected, undirected, and has distinct positive integer weights (i.e., all edge weights are different positive integers). We want to design an algorithm that updates the minimum spanning tree when a new edge $e = (u, v)$ is added to E, for some $u, v \in V$.

Alyssa P. Hacker suggests the following algorithm. To compute the minimum spanning tree T' of graph $G' = (V, E \cup \{e\})$ from T, we first add edge e to T, and then we delete the heaviest edge in the cycle that is created in T with the addition of edge e.

Prove that this algorithm constructs a minimum spanning tree for G'. **Note: this is asking for a proof of correctness, not alternate algorithms or runtime analyses.**

Fall 2014, Problem 2

(d) [*7 points]* After observing Alyssa's algorithm, Ben Bitdiddle wants to design an algorithm for updating the minimum spanning tree of a graph when new vertices are inserted.

Specifically, Ben wants an algorithm that takes:

- a minimum spanning tree T of some undirected graph $G = (V, E)$,
- v', a new vertex , and
- E', a set of new (weighted) edges between v' and V,

and computes a minimum spanning tree T' of the graph $G' = (V \cup \{v'\}, E \cup E')$. Once again, assume that all edge weights are distinct integers.

Design an $O(V \log V)$ time algorithm for Ben's problem.

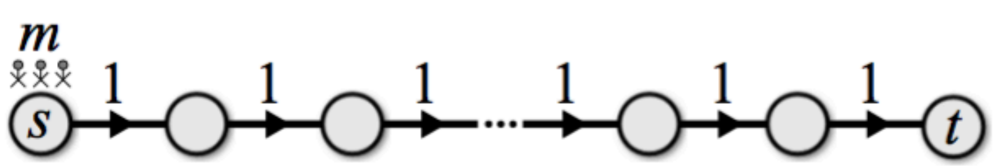
(e) *[**Bonus** 5 points]* Find an $O(V)$ algorithm for part (d). **This is a bonus problem, attempt it only if you have solved everything else.**

Spring 2015, Quiz 2, Problem 4

Problem 4. Maze Marathoner [20 points] (3 parts)

A group of m teens need to escape a maze, represented by a directed graph $G = (V, E)$. The teens all start at a common vertex $s \in V$, and all need to get to the single exit at $t \in V$. Every night, each teen can choose to remain where they are, or traverse an edge to a neighboring vertex (which takes exactly one night to traverse). However, each edge $e \in E$ has an associated capacity $c(e)$, meaning that at most $c(e)$ teens can traverse the edge during the same night. The goal is to minimize the number of nights required for all teens to escape by reaching the goal t.

(a) [3 points] First look at the special case where the maze is just a single path of length $|E|$ from s to t, and all the edges have capacity 1 (see below). Exactly how many nights are required for the teens to escape?



Spring 2015, Quiz 2, Problem 4

(b) [7 points] The general case is more complex. Assume for now that we have a "magic" algorithm that calculates whether the teens can all escape using $\leq k$ nights. The magic algorithm runs in polynomial time: $k^{\alpha} T(V, E, m)$ where $\alpha = O(1)$.

Give an algorithm to calculate the minimum number of nights to escape, by making calls to the magic algorithm. Analyze your time complexity in terms of V, E, m, α, and $T(V, E, m)$.

Spring 2015, Quiz 2, Problem 4

(c) [10 points] Now give the "magic" algorithm, and analyze its time complexity.
Hint: Transform the problem into a max-flow problem by constructing a graph $G' = (V', E')$ where $V' = \{(v, i) \mid v \in V, 0 \leq i \leq k\}$. What should E' be?

Spring 2015, Problem 4 [We did not actually have time to do this problem]

Problem 4. Amortized Analysis. *[15 points]* Design a data structure to maintain a set S of n distinct integers that supports the following two operations:

1. INSERT(x, S): insert integer x into S.
2. REMOVE-BOTTOM-HALF(S): remove the smallest $\lceil \frac{n}{2} \rceil$ integers from S.

Describe your algorithm and give the worse-case time complexity of the two operations. Then carry out an amortized analysis to make INSERT(x, S) run in amortized *O(1)* time, and REMOVE-BOTTOM-HALF(S) run in amortized 0 time.

Test your intuition/knowledge

(e) **T / F** Suppose that an undirected graph $G = (V, E)$ with positive edge weights has a minimum spanning tree of weight W. For all pairs of nodes $u, v \in V$ the shortest path between u and v has length less than or equal to W.

Test your intuition/knowledge

(e) **T / F** Suppose that an undirected graph $G = (V, E)$ with positive edge weights has a minimum spanning tree of weight W. For all pairs of nodes $u, v \in V$ the shortest path between u and v has length less than or equal to W.

True!

Test your intuition/knowledge

(d) T F [4 points]

In a connected, weighted graph, every lowest weight edge is always in *some* minimum spanning tree.

Test your intuition/knowledge

(d) T F [4 points]

In a connected, weighted graph, every lowest weight edge is always in *some* minimum spanning tree.

True!

Test your intuition/knowledge

(h) T F [4 points]
Every linear program has a unique optimal solution.

Test your intuition/knowledge

(h) **T** **F** [4 points]
Every linear program has a unique optimal solution.

False!

Test your intuition/knowledge

(d) Let $G = (V, E)$ and let H be the subgraph of G induced by some set of vertices $V' \subset V$. That is, $H = (V', E')$ where E' consists of all edges both of whose endpoints are in V'. Then every MST of H is a subgraph of some MST of G.

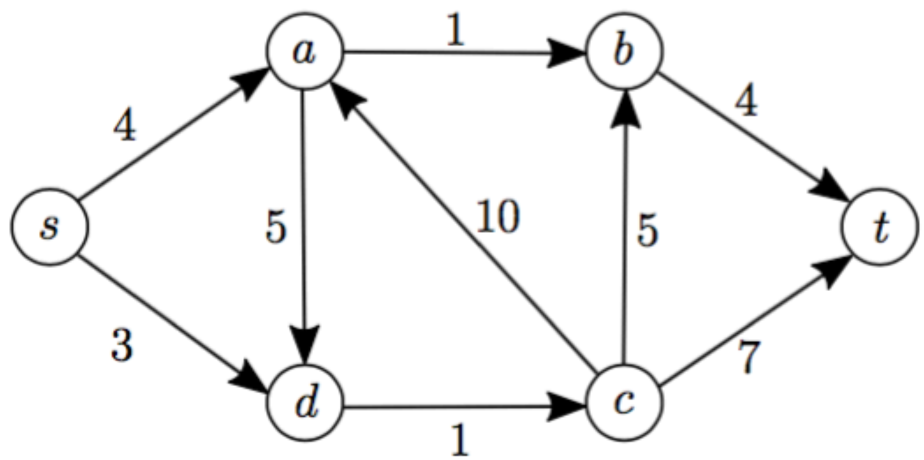
Test your intuition/knowledge

(d) Let $G = (V, E)$ and let H be the subgraph of G induced by some set of vertices $V' \subset V$. That is, $H = (V', E')$ where E' consists of all edges both of whose endpoints are in V'. Then every MST of H is a subgraph of some MST of G.

False!

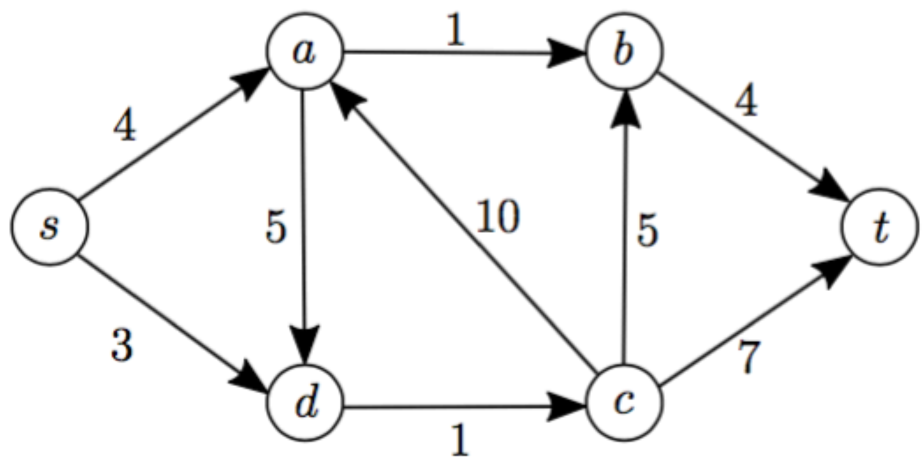
Test your intuition/knowledge

(a) [2 points] In the following network, it is possible to send 3 units of flow from node s to node t.



Test your intuition/knowledge

(a) [2 points] In the following network, it is possible to send 3 units of flow from node s to node t.



False!

Test your intuition/knowledge

(b) [4 points] If the primal Linear Program is feasible and has a bounded optimum, then the dual Linear Program is feasible.

Test your intuition/knowledge

(b) [4 points] If the primal Linear Program is feasible and has a bounded optimum, then the dual Linear Program is feasible.

True!

Test your intuition/knowledge

(d) [4 points] Let $G = (V, E)$ be a connected graph with unique edge weights where $|E| \geq 2$ with at most one edge between each pair of vertices. Then the edge with the second smallest weight must be in the MST for G.

Test your intuition/knowledge

(d) [4 points] Let $G = (V, E)$ be a connected graph with unique edge weights where $|E| \geq 2$ with at most one edge between each pair of vertices. Then the edge with the second smallest weight must be in the MST for G.

True!