CS 458, Data Structures and Algorithms II
Midterm

May 9, 2020

Due 4 hours after accessing the midterm on Blackboard. Submission on Gradescope
Last submission 11:59pm Wednesday May 13, 2020. Collaboration is not allowed

For all of the problems you need to provide a proof of correctness for your algorithm as well as a sound explanation for the runtime. For any graph related problem n will correspond to the number of vertices and m to the number of edges.

Problem 1: Counters (30 Points)

a . **(15 Points)** Consider adding a decrement operation to a binary counter. Provide a counter-example that is a series of n operations of INCREMENT or DECREMENT such the amortized cost per operation is $O(k)$, where k is maximum bit size.

b . **(15 Points)** Consider a ternary counter, a counter that uses the ternary numerical system (base 3). Prove that the operation INCREMENT using the ternary counter has amortized cost of $O(1)$.

Problem 2: All-Pairs Shortest Paths with Mandatory Vertices (35 Points)

Consider a graph $G = (V, E)$ where some vertices are colored blue. Let $B \subseteq V$ be the set of blue vertrices. Our goal is to compute all pairs shortest paths such that each shoretest path uses at least one blue vertex (if either i or j is blue then this immediately satisfies the constraint without posing any restrictions on the path).

a. **(15 Points)** Compute all-pairs shortest paths under this new constraint where there is exactly one blue vertex, i.e., $|B| = 1$. The running time of your algorithm should be $O(n^3)$.

b. **(20 Points)** Compute all-pairs shortest paths under this new constraint for an arbitrary number of blue vertices $|B| \geq 1$. The running time of your algorithm should be $O(n^3)$.

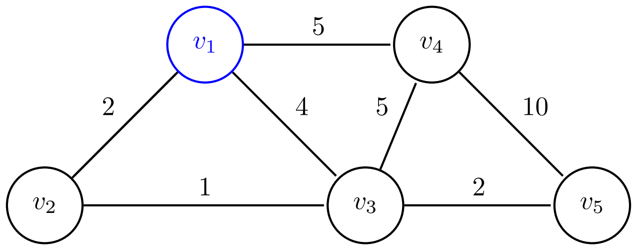

Figure 1: Without any restriction the shortest path between v_2 and v_4 is $v_2 \to v_3 \to v_4$ for a cost of 6. However, if the path needs to have one blue vertex then the optimal path is $v_2 \to v_1 \to v_4$ for a cost of 7.

Problem 3: (35 Points)

You decided to throw a costume party for all of your highschool friends, but here is a caveat, you want this to be a surprise and none should learn about the reunion before they arrive at your house. However, if any of your friends observes a familiar face dressed in a costume on their way to your house they will realize what is going on. To avoid this in your invitation letter you plan to provide directions for each of your friends how to arrive at your house in order to avoid this scenario.

More formally, you have an undirected graph G (the map of the city) in which vertices represent intersections and edges represent roads between them. The houses of your highschool friends are represented by special vertices $s_1, s_2, \ldots, s_k$ and your house is represented by an additional special vertex t. A suggsted route for your friend i corresponds to a path from s_i to t.

a. **(15 Points)** Give an efficient algorithm that determines whether or not all of your friends can arrive at your house using routes in the map such that no road is used by more than one of your friends.

b. **(20 Points)** You realized that your solution for Part a. had a terrible mistake. Even though all roads are used by at most one route that you designed for your friends they are still able to see each other while crossing the same intersection! Correct your solution for Part a. by giving an efficient algorithm that determines whether or not all of your friends can arrive at your house using routes in the map such that no road or intersection is used by more than one of your friends.