COMP285: Analysis of Algorithms
North Carolina A&T State University
Dr. Allison Sullivan

Homework #6
Due: October 12, 2018 (*in class quiz*)

Homework #6

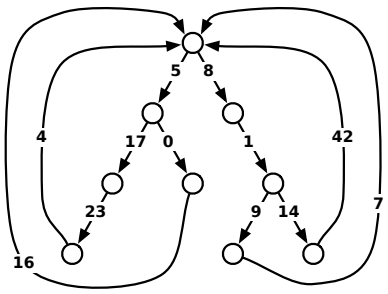
You should try to solve these problems by yourself. I recommend that you start early and get help in office hours if needed. If you find it helpful to discuss problems with other students, go for it. **You do not need to turn in these problems. The goal is to be ready for the in class quiz that will cover the same or similar problems.**

Problem 1: Single Source Shortest Paths

Recall Dijkstra's algorithm:

```
1  Dijkstra(G, w, s){
2     InitializeSingleSource(G, s)
3     S = {}
4     Q = G.V
5     while (Q != {}) {
6        u = Extract-Min(Q)
7        S = S + u
8        for each vertex v in Adj[u] {
9           Relax(u,v,w)
10       }
11    }
12 }
```

A *looped tree* is a weighted directed graph built from a binary tree by adding an edge from every leaf back to the root. Every edge has a non-negative weight. An example of a looped tree is shown below.



A looped tree.

(a) How much time would Dijkstra's algorithm require to compute the shortest path from u to v in a looped tree with n nodes? (Do NOT assume that either u or v is the root of the tree, though one could be.)

(b) Describe and analyze a faster algorithm to find the shortest path from u to v in a looped tree.

Problem 2: Optimizing Dijkstra's

Recall Dijkstra's algorithm:

```
1  Dijkstra(G, w, s){
2     InitializeSingleSource(G, s)
3     S = {}
4     Q = G.V
5     while (Q != {}) {
6        u = Extract-Min(Q)
7        S = S + u
8        for each vertex v in Adj[u] {
9           Relax(u,v,w)
10       }
11    }
12 }
```

Let $G = (V, E)$ be a weighted, directed graph with nonnegative weight edges, such that each weight is an integer in $\{0, 1, \ldots W\}$ for some non-negative integer W.

(a) Given this constraint on the inputs to the Djikstra's algorith, state the maximum possible cost of the shortest path from any node u to a node v.

(b) Describe how you can use this knowledge to create a more efficient data structure to select the next vertex to extract in Djikstra's algorithm.

Problem 3: Modifying Dijkstra's Algorithm

We are given an directed graph $G = (V, E)$ on which each edge $(u, v) \in E$ has an associated value $r(u, v)$, which is a real number in the range $0 \leq r(u, v) \leq 1$ that represents the reliability of a communication channel from vertex u to vertex v. We interpret $r(u, v)$ as the probability that the channel from u to v will not fail, and we assume that these probabilities are independent. Give an efficient algorithm to find the most reliable path between two given vertices.