CMP_SC 3050: Homework 4

This is a practice homework. Please do not hand in the homework. We shall discuss the answers in class on April 2, 2015.

1. Recall the Dijkstra algorithm we did in class (and given in Section 24.3 of the book). The Dijkstra algorithm computes the shortest distance from a source vertex to every other vertex in a weighted directed graph as long as the weights are non-negative. (By shortest, we mean that the sum of all weights of the edges should be minimum amongst all paths).

   (a) Consider the weighted directed graph G shown in Figure 1:

   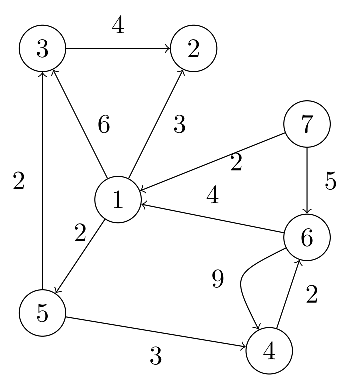

   Figure 1: Graph G

   For each vertex u in the graph, show how the values $u.d$ change when Dijkstra's algorithm is executed with the vertex numbered 1 as the source vertex.

   (b) Dijkstra algorithm is supposed to run only on graphs with non-negative weights. Let us see what happens when we run Dijkstra's algorithm on graphs with negative weights. Consider the weighted directed graph H shown in Figure 2:

   - What is the shortest distance from the vertex labeled s to all other vertices in the above graph (3 points)?
   - Suppose we run the Dijkstra's algorithm on the graph H with s as the source vertex. What are the values of the distances computed?

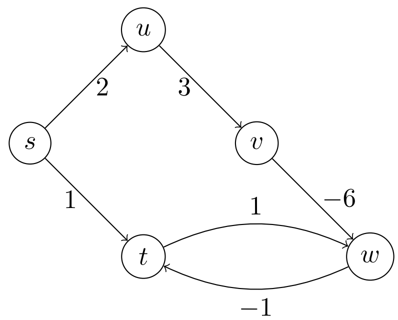

Figure 2: Graph H

2. A weighted undirected graph $G = (V, E)$ is an *almost-tree* if

    - G is connected, and
    - $|E| = |V|$ (that is the number of edges and the number of vertices are exactly the same).

    Give an algorithm that given an almost-tree $G = (V, E)$ computes the minimum spanning tree of G and runs in $O(|V|)$ time. You must justify why your algorithm works and runs in $O(|V|)$ time in order to get full credit.

3. A sequence of integers $A = a_1, a_2, \ldots, a_n$ is a subsequence of sequence $B = b_1, b_2, \ldots, b_m$ if $a_1, \ldots, a_n$ occur in B in the same order (may or may not occur next to each other). For example,

    - $A = 1, 2, 3, 4$ is a subsequence of $B = 6, 1, 2, 3, 4, 5$.
    - $A = 1, 2, 3, 4$ is **not** a subsequence of $B = 6, 1, 2, 4, 5$.
    - $A = 4, 2, 3, 1$ is a subsequence of $B = 6, 4, 2, 7, 3, 9, 1$.
    - $A = 4, 2, 3, 1$ is a subsequence $B = 3, 4, 2, 7, 3, 9, 4, 3, 1$.
    - $A = 4, 2, 3, 1, 6$ is **not** a subsequence of $B = 4, 2, 7, 3, 9, 6, 4, 3, 1$.

    Give an algorithm that given two sequences of integers A and B checks if A is a subsequence of B or not. Your algorithm must run in $O(n + m)$ time where n and m are the number of elements in the sequences A and B respectively. You must justify why your algorithm works and runs in $O(m + n)$ time in order to get full credit.

4. Recall the disjoint-set data structure implemented as a forest that we discussed in class and discussed in Section 21.3 of the book. We will assume that the data structure always uses union by rank and path compression heuristics. Assuming that we start a set of objects $x_1, x_2, \ldots x_{16}$, and perform the following sequence of operations:

```
1   for i = 1 to 16
2       Make-Set(x_i)
3   for i = 1 to 8
4       Union(x_{2i-1}, x_{2i})
5   for i = 1 to 4
6       Union(x_{4i-3}, x_{4i-1})
7   Union(x_1, x_5)
8   Union(x_11, x_13)
9   Union(x_1, x_10)
10  Find-Set(x_2)
11  Find-Set(x_9)
```

Draw the data structure after the lines 7,8,9,10 and 11 are executed.

5. You are given a list L of n natural numbers $d_1, d_2, \ldots, d_n$. Let $m = d_1 + d_2 + \cdots + d_n$. Give an $O(n \log n + m)$ algorithm that takes the list L and returns

   - true if there exists an <u>undirected</u> graph $G = (V, E)$ such that $|V| = n$ and for each number d_i in the list L, there is a vertex of G with degree exactly d_i;
   - and false otherwise.