CSE 101 Fall 2018
Homework 3
Due: Tuesday, October 23, 2018 at 11:59pm

INSTRUCTIONS

This homework assignment may be completed in groups of size 1-4. The solutions must be **typed** (using a computer.) Figures and graphs can be hand drawn. For algorithm descriptions we require a high-level English description AND an implementation description. If you find it necessary to also include a pseudo code to help understand your description then feel free to include it.

Please refer to the course page for requirements in writing up answers for algorithm questions.

---

1. For each type of graph, give a time analysis for Dijkstra's algorithm using an array for the priority queue and using a binary heap for the priority queue. Which one is more efficient in each case?
(6.25 points each.)
   (a) A complete graph on n vertices
   (b) A complete bipartite graph on two sets each with $n/2$ vertices.
   (c) A $\sqrt{n} \times \sqrt{n}$ grid.
   (d) The hypercube of size $n = 2^k$. (A hypercube of size 2^k is a graph where vertices are binary strings of length k and there is an edge from b_1 to b_2 if they differ by one bit.)

2. Say you have k sorted lists, each with n/k elements. Show how to use a heap to merge all of these lists into a single sorted list in $O(n \log k)$ time.
(11 points correct algorithm, 7 points for proof of correctness, 7 points for time analysis and efficiency.)

3. The following statements may or may not be correct. In each case, either prove it (if it is correct) or give a counterexample (if it is not correct). Always assume that the graph $G = (V, E)$ is undirected, connected and the edge weights are positive. (5 points each)
   (a) If a graph G has more than $|V| - 1$ edges, and there is a unique heaviest edge, then this edge cannot be part of a MST.
   (b) If G has a cycle with a unique heaviest edge e, then e cannot be part of an MST.
   (c) Let e be any edge of minimum weight in G. Then e must be part of some MST.
   (d) If the lightest edge in a graph is unique, then it must be part of every MST.
   (e) The shortest path tree computed by Dijkstra's is necessarily an MST.

4. Often there are multiple shortest paths between two nodes of a graph. Give a linear-time algorithm for the following task.

   Input: Undirected graph $G = (V, E)$ with unit edge lengths; nodes $u, v \in V$.
   Output: The number of distinct shortest paths from u to v.

   (13 points for algorithm description, 6 points proof of correctness 6 points time analysis, space analysis and efficiency.)