MAS 714, Fall 2019
Tutorial 3/Homework 1

This homework will be graded. Please hand the solutions in before Monday, September 9, 11:30 am.
Solutions can be submitted either via email of in written form.

Problem 1 We are given a directed graph $V = (V, E)$ with weights $W(u, v) \in [0, 1]$. $W(u, v)$ is the probability that the edge (u, v) fails. All failure probabilities are independent, i.e., the probability that edges (u, v) and (a, b) both fail is $W(u, v) \cdot W(a, b)$. Describe an efficient algorithm that, given vertices s, t in G, finds a path from s to t with least total failure probability, i.e., a path for which the probability that any edge fails is minimal among all s, t paths.

Problem 2 We are given a graph G with vertex weights $W(v)$ for all $v \in V$. For a vertex $v \in V$ define $\min(v)$ to be the minimum $W(u)$ over all vertices u that are reachable from v in G, i.e., for which there is a path from v to u. Describe an algorithm with time complexity $O(n + m)$ that computes $\min(v)$ for all vertices v of G.

Problem 3 If we build a heap of n elements using n Insert operations, the time we use is $O(n \log n)$. Show how to build a heap of n elements in time $O(n)$.

Problem 4 A d-ary heap stores elements in an array such that their order corresponds to a tree where each vertex (except leaves) has exactly d children (except the rightmost vertex on the second lowest level that can have between 1 and d children). The heap property for d-ary heaps still says that the key of a parent node must be less or equal than the keys of child nodes.
(a) Informally describe procedures for ExtractMin,DecreaseKey, and Heapify, and analyze their running time in terms of n and d.
(b) A graph is ϵ-dense, if it has at least $m \geq n^{1+\epsilon}$ edges, for some constant $\epsilon > 0$. Describe a priority queue implementation, such that Dijkstra (using that priority queue) will run in time $O(m)$ for ϵ-dense graphs.

Problem 5 A graph is given as an adjacency list with edge weights, where all edge weights are from $\{0, \ldots, T\}$. Describe an algorithm that computes shortest paths from a vertex s in time $O(Tn + m)$.