Problem 1

Let $R$ be an integer between 1 and $n$. Modify Dijkstra’s algorithm to design an $O(R^2 \log n)$ time algorithm that given any $n$-node graph $G = (V,E)$ with nonnegative integer edge weights $w : E \to \mathbb{Z}^+$, and a source $s$, finds the closest $R$ vertices $T_s$ to $s$, and the distance between $s$ and every $v \in T_s$, under the following assumptions:

1. There is a data structure $F$ (e.g. Binomial heap) that stores up to $n$ pairs $(e,k)$ (where $e$ is an element and $k$ is an integer value) and supports the following operations each in $O(\log n)$ time:
   - insert($e,k$): insert an element $e$ into $F$ with value $k$, provided $e$ is not in $F$ yet with any value
   - decrease-key($e,k$): if $e$ is in $F$ with value $k' \geq k$, change its value to $k$
   - extract-min: return $(e,k)$ where $e$ has the minimum value $k$ over all elements in $F$, deleting $(e,k)$ from $F$

2. $G$ is given in adjacency list representation, and for each $u \in V$, the neighbors of $u$, $N(u)$ are sorted in nondecreasing order of their edge weight.

Give pseudocode for your algorithm, prove that it is correct and that it runs in $O(R^2 \log n)$ time.

Problem 2

The radius of a graph is given by $R = \min_s \max_u d(u,v)$. In this problem we will adapt the diameter approximation algorithm given in class to obtain an $\tilde{O}(m \sqrt{n})$ time 3/2-approximation algorithm for the radius $R$ of any given undirected graph on $n$ nodes and $m$ edges, whenever $R$ is even.

The eccentricity $\epsilon(v)$ of a node $v$ is defined as the maximum distance from $v$ to another node, i.e. $\epsilon(v) := \max_{u \in V} d(u,v)$.

The center $c$ of a graph $G$ is the node in $G$ of minimum eccentricity, i.e. $c := \arg \min_{v \in V} \epsilon(v)$.

Assume below that the radius of the given graph $G$ is even. Let $S$ be a random sample of $O(\sqrt{n} \log n)$ nodes, let $w$ be the node furthest from $S$ and $T_w$ be the closest $\sqrt{n}$ nodes to $w$, just as in the diameter algorithm from class. You can assume that $S$ hits $T_w$, as we showed in class that it will do so with high probability.

- Show that if for some node $s$ in the random sample $S$, $d(s,c) \leq R/2$, then $R \leq \min_{s \in S} \epsilon(s) \leq 3R/2$, and hence one can return an estimate $R'$ of the radius so that $R \leq R' \leq 3R/2$.
- Show that if for all nodes $s \in S$, $d(s,c) > R/2$, then all nodes at distance $R/2$ from $w$ are in $T_w$.
- Show that if $d(w,c) \leq R/2$, then $R \leq \epsilon(w) \leq 3R/2$.

• Show that if $d(w,c) > R/2$ and for all nodes $s \in S$, $d(s,c) > R/2$, then there is some node $x$ in $T_w$ with $\epsilon(x) \leq 3R/2$, and hence $R \leq \min_{x \in T_w} \epsilon(x) \leq 3R/2$.

- Give pseudocode for the radius approximation algorithm.
Problem 3

Design an $O(n^{2.5} \log n)$ time algorithm that, given a graph with nonnegative edge weights, finds with high probability, the distances between all pairs of vertices for which there is a shortest path that uses at least $\sqrt{n}$ nodes.