1. Let $G$ be a directed graph with (possibly negative!) edge weights, and let $s$ be an arbitrary vertex of $G$. Suppose every vertex $v \neq s$ stores a pointer $\text{pred}(v)$ to another vertex in $G$.

Describe and analyze an algorithm to determine whether these predecessor pointers define a single-source shortest path tree rooted at $s$. Do not assume that the graph $G$ has no negative cycles.

[Hint: There is a similar problem in head-banging, where you’re given distances instead of predecessor pointers.]

2. Let $G$ be a directed graph with positive edge weights, and let $s$ and $t$ be an arbitrary vertices of $G$. Describe an algorithm to determine the number of different shortest paths in $G$ from $s$ to $t$. Assume that you can perform arbitrary arithmetic operations in $O(1)$ time. [Hint: Which edges of $G$ belong to shortest paths from $s$ to $t$?]

3. Describe and analyze and algorithm to find the second smallest spanning tree of a given undirected graph $G$ with weighted edges, that is, the spanning tree of $G$ with smallest total weight except for the minimum spanning tree.
1. Describe and analyze an algorithm to determine whether a given set of predecessor pointers define a single-source shortest path tree rooted at $s$. 
2. Describe an algorithm to determine the number of different shortest paths in a given directed graph from one given vertex $s$ to another given vertex $t$. 
3. Describe and analyze an algorithm to find the second smallest spanning tree of a given undirected graph with weighted edges.