Problems

1) The Bellman-Ford algorithm finds the shortest path from a source to all the vertices in the graph (if there are negative edges but no negative cycles present). The algorithm takes time $O(mn)$. But if we have a graph that’s a DAG, then we can do much better. Find a linear time algorithm for the following problem and prove that it works.

Input: A DAG $G = (V, E)$ with possibly negative edge lengths and vertex $s \in V$.
Output: An array $dist(\cdot)$, where $dist(u)$ is set to the shortest path length from $s$ to $u$.
(Hint: First use topological sort)

2) Suppose that in addition to having edge lengths, a graph also has vertex costs. The price of a path is defined as the sum of all the edge length and the vertex costs on the path (including the endpoints). Give an efficient algorithm for the following problem:

Input: A directed graph $G = (V, E)$ with positive edge lengths and vertex costs and a starting vertex $s$.
Output: An array $cost(\cdot)$ such that for every vertex $u$, $cost(u)$ is the cost of the cheapest path from $s$ to $u$.
Notice that $cost(s) = c_s$ where $c_s$ is the cost of vertex $s$.

3) Same as problem 2, except now assume that $G$ is an undirected graph.