Problem 1*. Prove the correctness of Dijkstra’s algorithm (when the edges have non-negative weights).

Problem 2. Consider again your proof for Problem 1. Point out the place that requires edge weights to be non-negative.

Problem 3* (SSSP in a DAG). Consider a simple acyclic directed graph $G = (V, E)$ where each edge $e \in E$ has an arbitrary weight $w(e)$ (which can be negative). Solve the SSSP problem on $G$ in $O(|V| + |E|)$ time.

Problem 4. Let $G = (V, E)$ be a simple directed graph where each edge $e \in E$ carries a weight $w(e)$, which can be negative. It is guaranteed that $G$ has no negative cycles. Prove: given any vertices $s, t \in V$, at least one shortest path from $s$ to $t$ is a simple path (i.e., no vertex appears twice on the path).

Problem 5**. Let $G = (V, E)$ be a simple directed graph where the weight of an edge $(u, v)$ is $w(u, v)$. Prove: the following algorithm correctly decides whether $G$ has a negative cycle.

```
algorithm negative-cycle-detection
1. pick an arbitrary vertex $s \in V$
2. set $\lambda$ to the sum of all the positive edge weights in $G$
3. initialize $dist(s) = 0$ and $dist(v) = \lambda$ for every other vertex $v \in V$
4. for $i = 1$ to $|V| - 1$
5.    relax all the edges in $E$
6. for each edge $(u, v) \in E$
7.    if $dist(v) > dist(u) + w(u, v)$ then
8.        return “there is a negative cycle”
9.    return “no negative cycles”
```