1. (40 points) In class I mentioned that Dijkstra’s algorithm does not work when the edge lengths can be negative. Show this is possible in the strongest possible sense by giving an example that exhibits this. In particular, come up with an example of a directed graph (with no directed cycles) along with edge lengths (exactly one of which is negative) and two nodes $s$ and $t$ in the graph such that when Dijkstra’s algorithm is run on the graph, it does not output the shortest path between $s$ and $t$. (As before, the length of a path is the sum of the lengths of the edges in it.)

Note: You have the full freedom in picking the graph.

Give a proper example: 40 pts (this question is all or nothing. If your example is incorrect you get 0 points. If it’s correct, you get 40 points. No need for an argument of correctness, though you should convince yourself it’s correct by running Dijkstra’s algorithm on your graph.)

2. (45 points) Mr. Greedy Barber has to schedule $n$ clients for haircuts for the next day. These $n$ clients are regular customers, so for each customer $i$, Mr. Barber know the exact time $h_i$ it would take him to cut $i$’s hair. Mr. Barber is also gunning for a record, so he has decided not to waste any time moving from one haircut to the next. Help Mr. Barber design an $O(n \log n)$ time algorithm that schedules all the $n$ clients, which provably minimizes the sum of the service time for every client, where the service time for client $i$ is $h_i$ plus the time $i$ had to wait before $i$’s turn came by. You should assume that all the clients came into the shop at the same time. As usual, you must prove the optimality of your algorithm.

Here is a simple example: say $n = 3$ and
\[ h_1 = 5, \quad h_2 = 10, \quad h_3 = 4. \]

Now consider the schedule
\[ 1, 2, 3 \]

i.e. 1 get the haircut first and then leaves, 2 gets his haircut after 1 and 3 gets his hair cut after 2. Note that the service time for 1 is 5. The service time for client 2 is his wait time (which is 5) and $h_2$, which is 15. The service time for client 3 is his wait time (which is $5 + 10 = 15$) plus $h_3$, which is 19. Thus, the sum of the service times for this schedule is $5 + 15 + 19 = 39$. This schedule, however, is not optimal.

Correct algorithm: 10 pts

Proof that your algorithm is correct: 25 pts

Runtime analysis: 10 pts
3. (15 points) **Interview Question**

Let $T$ be a rooted tree with $n$ nodes. Each node $u$ has a nonnegative weight $w(u)$. The weight of a path is the sum of the weights of the nodes on the path.

Your goal is to efficiently assign a new weight $w'(u)$ to each node $u$ that satisfies the following conditions:

(a) Each path from the root to a leaf has the same weight $W^*$.

(b) For all nodes $u$, $w'(u) \geq w(u)$. (i.e. you can only increase the node weights.)

(c) $\sum_{u \in \text{nodes}(T)} w'(u)$ is minimized. (minimize the sum of all weights.)

Correct algorithm: 15 pts (no proof or runtime analysis required, though your algorithm must run in time polynomial in $n$.)