At the beginning of the trip, we know the distance from each gas station to the start of the route and the price of gas at that station (the distances and prices are unique, i.e. there are no ties). There is a gas station at the start point. We assume that the gas consumption of the vehicle is 1 unit of gas per mile. We know how long the route is; at the beginning of the trip, the gas tank is empty, and at the end of the trip, the tank should be empty again.

1. A greedy algorithm that solves this problem is as follows:

   Let $\mathit{milesTraveled} = 0$. Let *bestPriceSoFar* be the price of gas at the station at the start point. Let D be the total distance of the route. Let $\mathit{moneySpent} = \$0$. Create a queue Q of ordered pairs (d, p) representing gas stations where d is the distance to that station from the start point and p is the price of gas at that station. Put all gas stations, in increasing order of distance, into the queue, except for the gas station at the start point.

   While Q is not empty, remove the first (i.e. the next closest) station (d, p) from Q for consideration.

   > If $p < \mathit{bestPriceSoFar}$ then buy $d - \mathit{milesTraveled}$ units of gas from the current gas station, each unit of which will cost *bestPriceSoFar*, and increase *moneySpent* appropriately. Now drive $d - \mathit{milesTraveled}$ miles until you reach station (d, p). Update variables by letting $\mathit{bestPriceSoFar} = p$, $\mathit{milesTraveled} = d$.
   >
   > If $p > \mathit{bestPriceSoFar}$ then discard (d, p).

   When Q is empty, buy $D - \mathit{milesTraveled}$ units of gas at the current gas station, each unit of which will cost *bestPriceSoFar*. Drive to the end of the route.

   For example, suppose that the route is $D = 100$ miles. The gas station at the start point sells gas for \$4.50. Suppose that the queue is as follows:

(10, \$3)	(45, \$5)	(60, \$2.50)

   Now, *milesTraveled* is 0, *bestPriceSoFar* is \$4.50. We consider the first station in the queue, $(10, \$3)$. Since $p = \$3 < \$4.50$, then we'll buy $d - \mathit{milesTraveled} = 10 - 0 = 10$ units of gas at the current station, where the price is $\mathit{bestPriceSoFar} = \$4.50$. We then drive 10 miles to reach gas station $(10, \$3)$, and update variables: now $\mathit{milesTraveled} = 10$, $\mathit{bestPriceSoFar} = \$3$, and $\mathit{moneySpent} = \$45$.

   We then consider the next station in the queue, $(45, \$5)$. Since $\$5 > \mathit{bestPriceSoFar} = \$3$, we discard this station.

   We then consider the last station in the queue, $(60, \$2.50)$. Since $\$2.50 < \mathit{bestPriceSoFar} = \$3$, then we'll buy $d - \mathit{milesTraveled} = 60 - 10 = 50$ units of gas at the current station where the price is $\mathit{bestPriceSoFar} = \$3$. We then drive 50 miles to reach gas station $(60, \$2.50)$, and update variables: now $\mathit{milesTraveled} = 60$, $\mathit{bestPriceSoFar} = \$2.50$, and $\mathit{moneySpent} = \$195$.

   Now the queue is empty. We'll therefore buy $D - \mathit{milesTraveled} = 100 - 60 = 40$ units of gas at the current station where the price is $\mathit{bestPriceSoFar} = \$2.50$. Our final $\mathit{moneySpent} = \$295$. We then drive 40 miles to reach the end of the route. We note that at the end of the route, our tank is empty, as desired.

2. Now we want to prove that this algorithm is optimal.

   Proof. We can represent any feasible solution as a list of length D where each element in the list is the price paid for the unit of gas that took us that mile. For the example above, the solution given by the algorithm is

   $$\{\$4.50_1, \ldots, \$4.50_{10}, \$3_{11}, \ldots, \$3_{60}, \$2.50_{61}, \ldots, \$2.50_{100}\}$$

Let $A = \{a_1, \ldots, a_D\}$ be the solution given by the algorithm. Let $O = \{o_1, \ldots, o_D\}$ be the optimal solution. Now, we know that we must buy gas at the station at the start of the route, since we originally have no gas. Thus, we must have $a_1 = o_1$.

Now we know that $a_i = o_i$ for some i. Note by the construction of the algorithm, we are always buying just enough gas to get to a station where we can buy even cheaper gas; i.e. every unit of gas is bought at the lowest possible price.

Now suppose that $a_{i+1} \neq o_{i+1}$. By the construction of the algorithm, it is impossible that $a_{i+1} > o_{i+1}$: that would imply that we bought the o_{i+1} unit of gas for a lower price than we bought the a_{i+1} unit of gas, which is impossible because we always buy gas at the lowest possible price. Therefore, if $a_{i+1} \neq o_{i+1}$, then it must be that $a_{i+1} < o_{i+1}$. Then consider the following set O':

$$O' = O \backslash \{o_{i+1}\} \cup \{a_{i+1}\} = \{o_1, \ldots, o_i, a_{i+1}, o_{i+2}, \ldots, o_D\}$$

i.e. we replace o_{i+1} with a_{i+1} in O. But since $a_{i+1} < o_{i+1}$, it follows that

$$\sum_{o_j \in O'} o_j < \sum_{o_j \in O} o_j,$$

that is, the sum of the prices of units of gas in the O' solution is strictly less than the sum of the prices of units of gas in the O solution. This is a contradiction to the optimality of O. Therefore it is not possible that $a_{i+1} \neq o_{i+1}$ for any i. By induction, we get that $a_i = o_i$ for all i, and thus $A = O$. Our algorithm's solution is equal to the optimal solution; our algorithm is optimal. □

3. Finally we analyze the time complexity. First we need to sort the gas stations in order of increasing distance from the start point, which takes $O(n \log n)$. Then we consider each gas station, one at a time, and perform $O(1)$ operations (comparing prices) on each, in total $O(n)$. The total time complexity is $O(n \log n) + O(n) = O(n \log n)$.