Package Delivery

Calvin Fung

August 17, 2025

1 Problem

Johnny drives a truck and must deliver a package from his hometown to the district center. His hometown is located at point 0 on a number line, and the district center is located at the point d.

Johnny's truck has a gas tank that holds exactly n liters, and his tank is initially full. As he drives, the truck consumes exactly one liter per unit distance traveled. Moreover, there are m gas stations located at various points along the way to the district center. The i-th station is located at the point x_i on the number line and sells an unlimited amount of fuel at a price of p_i dollars per liter. Find the minimum cost Johnny must pay for fuel to successfully complete the delivery.

It's given $1 \leq x_i \leq d-1, 1 \leq p_i$ and that the positions of the gas stations are distinct.

2 Solution

We present an $O(m \lg m)$ time greedy solution.

2.1 analysis

Suppose the stations are sorted in increasing distance from hometown. For a station k, Define

- The n window from k, $W_k = \{i = k+1, ..., m : x_i - x_k \leq n\}$
- The next station that's cheaper then k, $s_k \in \{k+1, ..., m\}$ which is the smallest index such that $p_{s_k} < p_k$.

We adopt a greedy strategy, given we need to refill at station k, the greedy choice for the next station that we refill at is :

case 1 : The district center is very close $d - x_k \leq n$. If s_k exist, refill at s_k, else let k be the last station we refill at.

case 2 : The district center is far $d - x_k > n$. If $s_k \in W_k$, refill at s_k, else refill at the cheapest station in W_k (if there're multiple of them, refill at the one with the smallest index).

It's easy to see that if we're in case 2 and $W_k = \emptyset$, then Johnny cannot successfully complete the delivery.

Notice once the next station to be refilled is decided, the amount to refill at the current station can also be greedily decided : If the next station is cheaper (or has the same price), then refill the least amount at the current station ie. just enough to get to the next station. Else, refill as much as possible at the current station ie. full tank. It's not hard to see the correctness.

For the base case, it's very simple and correctness follows from that of above by letting the hometown to be a station with price 0.

case 1 : The district center is very close $d \leq n$. Then just drive there with cost 0. case 2 : The district center is far $d > n$. Then s_0 doesn't exist so refill at the cheapest station in W_0 (if there're multiple of them, refill at the one with the smallest index).

Below we prove our greedy strategy is correct. Let G, O be greedy and optimal solution respectively. Consider the sequence of stations where we refill at, suppose G, O agree until station k and Johnny has the same amount of gas g when arriving at k in both G, O.

case 1a : The district center is very close $d - x_k \leq n$ and s_k exists (notice $s_k \in W_k$). We can assume $x_{s_k} - x_k > g$ otherwise O will skip station k.

If k is the last station in O then it will refill $d - x_k - g$ amount of gas at k. The associated cost is $(d - x_k - g)p_k$. Now consider an alternative, refilling $x_{s_k} - x_k - g$ amount of gas at k and $d - x_{s_k}$ at s_k. Notice this is feasible and we can reach district center with this strategy, the cost is $(x_{s_k} - x_k - g)p_k + (d - x_{s_k})p_{s_k} < (x_{s_k} - x_k - g)p_k + (d - x_{s_k})p_k = (d - x_k - g)p_k$ a contradiction that O is optimal.

Below suppose the optimal refills $f(i)$ amount of gas at i.

If the next station that O refills at is $w > s_k$ where it refills $f(w)$ amount of gas. When leaving w Johnny has $g + f(k) + f(w) - x_w + x_k$ gas and cost is $f(k)p_k + f(w)p_w$. Now consider an alternative, refilling $x_{s_k} - x_k - g$ amount of gas at k, $f(k) - (x_{s_k} - x_k - g)$ at s_k and $f(w)$ at w. Notice this's feasible and when leaving w Johnny has $g + f(k) + f(w) - x_w + x_k$ gas and cost is $(x_{s_k} - x_k - g)p_k + (f(k) - (x_{s_k} - x_k - g))p_{s_k} + f(w)p_w < (x_{s_k} - x_k - g)p_k + (f(k) - (x_{s_k} - x_k - g))p_k + f(w)p_w = f(k)p_k + f(w)p_w$ a contradiction that O is optimal. [1]

If O refills at a non empty $W \subseteq \{i = k+1, ..., s_k\} \setminus \{s_k\}$. When leaving s_k Johnny has $g + f(k) + f(s_k) + \sum_{w \in W} f(w) - s_k + x_k$ gas and cost is $f(k)p_k + f(s_k)p_{s_k} + \sum_{w \in W} f(w)p_w$. Now consider an alternative, refilling $x_{s_k} - x_k - g$ amount of gas at k, $f(k) + f(s_k) + \sum_{w \in W} f(w) - (x_{s_k} - x_k - g)$ at s_k. Notice this's feasible and when leaving s_k Johnny has $g + f(k) + f(s_k) + \sum_{w \in W} f(w) - s_k + x_k$ gas and cost is

$$
\begin{aligned}
&(x_{s_k} - x_k - g)p_k + (f(k) + f(s_k) + \sum_{w \in W} f(w) - (x_{s_k} - x_k - g))p_{s_k} \\
&< (x_{s_k} - x_k - g)p_k + f(k)p_k + f(s_k)p_{s_k} + \sum_{w \in W} f(w)p_w - (x_{s_k} - x_k - g)p_k \\
&= f(k)p_k + f(s_k)p_{s_k} + \sum_{w \in W} f(w)p_w
\end{aligned}
$$

a contradiction that O is optimal. [2]

Therefore, in this case, the next station that O picks must be s_k which is our the greedy choice.

case 1b : The district center is very close $d - x_k \leq n$ and s_k doesn't exist. We can assume $d - x_k > g$ otherwise O will skip station k.

If O refills at a non empty $W \subseteq \{i = k+1, ..., m\}$. The cost is $f(k)p_k + \sum_{w \in W} f(w)p_w$. Now consider our greedy choice, it refills $d - x_k - g$ amount of gas at k and skip all remaining stations. Notice this's feasible with cost $(d - x_k - g)p_k \leq f(k)p_k + \sum_{w \in W} f(w)p_w$ because $p_k \leq p_w, \forall w \in W$ and $g + f(k) + \sum_{w \in W} f(w) \geq d - x_k$. Therefore, in this case if O doesn't make the greedy choice, we can modify so that it makes the greedy choice and obtain a solution that's no worse.

case 2a : The district center is far $d - x_k > n$ and $s_k \in W_k$. We can assume $s_k - x_k > g$ otherwise O will skip station k.

Notice O must refill at least a station in W_k otherwise it cannot reach d. It follows O must make the greedy choice and the proof is the same as [1],[2] above.

case 2b : The district center is far $d - x_k > n$ and either s_k doesn't exist or $s_k \notin W_k$. Let σ be cheapest station in W_k (if there're multiple of them, be the one with the smallest index).

Notice O must refill at least a station in W_k otherwise it cannot reach d. It follows if O doesn't make the greedy choice we can modify so that it does and obtain a solution that's no worse, the proof is similar to [1],[2] above, just replace s_k by σ and change $<$ to $\leq$.

2.2 algorithm

We present an $O(m \lg m)$ time solution. It implements above greedy strategy with monotonic stack. Run time is dominated by the initial sorting, other parts are linear time.

```
//G are the gas stations. Assume the hometown G[0].x = 0, G[0].p = 0
f(d,m,n,G){
        //easy base case
        if(d <= n)      return 0;

        sort G in increasing x value

        //S[k] = s_k; W[k] = cheapest station in W_k with smallest index
        S,W = arrays of length m
        //monotonic stack
        M = empty stack

        /*
        Compute S,W
        notice W[k] maybe wrong ie. when it exists but the while loop doesn't run
        because G[M.top()].p < G[k].p,
        notice G[M.top()] must be the station right after G[k] so in this case
        G[M.top()] = s_k, if it's in W_k then we don't need W[k] else
        Johnny cannot successfully complete the delivery so we also don't need W[k],
        either way we're fine
        */
        for(k = m;k>=0;--k){
                V = inf
                W[k] = -1
                while(!M.empty() && G[M.top()].p >= G[k].p){
                        //update W[k] ?
                        if(G[M.top()].x - G[k].x <= n && G[M.top()].p < V){
                                W[k] = M.top()
                                V = G[M.top()].p
                                M.pop()
                        }
                }
                S[k] = -1
                if(!M.empty()){
                                S[k] = M.top()
                }
                M.push(k)
        }

        //greedy
        k = 0 //current position
        g = n //current amount of gas
        opt = 0 //current cost
        while(1){
                if(d - G[k].x <= n){

                        if(S[k] != -1){
                                opt += (G[S[k]].x - G[k].x - g)*G[k].p //just enough
                                g = 0 //amount of gas when Johnny arrives at S[k]
                                k = S[k]
```

```
                }else{
                        opt += (d - G[k].x - g)*G[k].p //just enough
                        break;
                }

        }else {
                //s_k ∈ W_k ?
                if(S[k] != -1 && G[S[k]].x - G[k].x <= n){
                        opt +=  (G[S[k]].x - G[k].x - g)*G[k].p //just enough
                        g = 0 //amount of gas when Johnny arrives at S[k]
                        k = S[k]
                }
                //W_k = ∅ ?
                else if (W[k] != -1){
                        if(G[W[k]].p > G[k].p){
                                opt +=  (n - g)*G[k].p //go full tank
                                //amount of gas when Johnny arrives at W[k]
                                g = n - (G[W[k]].x - G[k].x)
                        }else{
                                opt +=  (G[W[k]].x - G[k].x - g)*G[k].p //just enough
                                //amount of gas when Johnny arrives at W[k]
                                g = 0
                        }
                        k = C[k]
                }else{
                        return -1;
                }
        }
    }

    return opt;

}
```