Question 2

There are n towns in a line. A traveller starts at town 1 and wants to reach town n. It takes a week to travel from any town to the next.

In each town, there is a market where you can buy rations. In the market of town i, you can buy a week's worth of rations for c_i dollars. Furthermore, you can buy several weeks' rations for later consumption.

Your goal is to calculate the minimum cost to travel from town 1 to town n.

2.1 **[3 marks]** Which town's rations do you want to consume as you travel from town $n-1$ to town n? Provide reasoning to support your answer.

Answer:

We hope to buy from the place with the lowest ration price before $n-1$.

Suppose P is the array of ration price, i is the town's position, n is the number of towns, $P[i]$ is the ith town's ration price.

- If the minimum price during $[1, n-1) < P[n-1]$, buy ration in the town with the minimum price.
- If the minimum price during $[1, n-1) = P[n-1]$, buy ration in the town with the minimum price or in $(n-1)th$ town.
- If the minimum price during $[1, n-1) > P[n-1]$, buy ration in $(n-1)th$ town.

2.2 **[3 marks]** When is the last time (if any) that you should change using from one town's rations to another? Provide reasoning to support your answer.

Answer:

Suppose P is the array of ration price, i is the town's position, $P[i]$ is the ith town's ration price, n is the number of towns, the last time to buy is in mth town.

We hope thaSuppose P is the array of ration price, i is the town's position, $P[i]$ is the ith town's ration price, the last time to buy is in mth town.

$P[i]_{min} = P[m]$ is the minimum price in array P.

Because, when we find the least price during all towns, we can buy $n-m-1$ number of ration. $P[m]$ is the minimum price during all towns, therefore, $P[m] \times (n-m-1)$ is the minimum price which we paid after than mth towns.

2.3 **[14 marks]** Design an algorithm which runs in $O(n)$ time and achieves the goal.

Answer:

Suppose P is the array of ration price, N is the number of ration purchased, i is the town's position,m record the minimum position during the array P, n is the number of towns

First, we must buy ration in the first town because we do not carry any ration during the traveling, set $m = 1$ and $N[1] = 1$.

Because when we reach nth town, the traveling is finished, therefore, we do not need to buy in the last town.
Now, perform the following step while $i > 1 \, \text{AND} \, i < n$:

- if $P[i] >= P[m]$, $N[m] = N[m] + 1$ and continue.
- if $P[i] < P[m]$, $N[i] = 1$ and $m = i$ than continue.

The time complexity is $O(n)$

After that, traverse the array N again, the time complexity is $O(n)$ and we can get the number of ration which need to buy in each towns, the minimum total price is the sum value of array N.

Therefore, the total time complexity is $O(n)$.