3 Radio Towers

Written by: **Kevin Luiz Ponte Pucci** (Portugal), Oporto University
Prepared by: Jonathan Irvin Gunawan
Solutions, review, and other problem preparations by: Abdul Malik Nurrokhman, Muhammad Ayaz Dzulfikar, Prabowo Djonatan
Analysis author: Jonathan Irvin Gunawan

3.1 Subtask 1

In this subtask, the height of the towers is a bitonic sequence.

Let k be the index such that $H[k-1] < H[k] > H[k+1]$. It is impossible to lease two towers, both with indices not more than k. It is also impossible to lease two towers, both with indices not less than k.

If $R \leq k$ or $L \geq k$, then there is no way to lease more than one tower. Otherwise, we check whether we can lease both tower L and tower R using tower k as an intermediary.

Time complexity: $O(N+Q)$

3.2 Subtask 2

In this subtask, $Q = 1, N \leq 2000$.

Observation 3.1. *We can greedily try to lease towers from the lowest height.*

Proof. We can prove this by contradiction. Suppose tower x is the tower of the lowest height that the greedy solution can lease, but there is a more optimal solution that does not lease building x. Let a and b be the nearest tower to the left and right of tower x that is higher than tower x and is leased in the optimal solution. If the intermediary tower for towers a and b is to the left of tower x, then we can lease tower x and not lease tower b instead. Similarly, if the intermediary tower for towers a and b is to the right of tower x, then we can lease tower x and not lease tower a instead. This means that the greedy solution will not produce a less optimal solution. □

Definition 3.1. *For each tower x, $L[x]$ (and $R[x]$) is the nearest tower to the left (and to the right) of tower x with the height of at least $H[x] + \delta$.*

We iterate towers from the lowest height, and we can lease tower x if and only if there is no previously leased tower between towers $L[x]$ and $R[x]$. A naive implementation of this solution runs in $O(QN^2)$.

Time complexity: $O(QN^2)$

3.3 Subtask 3

In this subtask, $Q = 1$.

To solve this subtask, for each x, we can find $L[x]$ and $R[x]$ in $O(\log N)$ using segment tree. We can also get the minimum tower height between towers $L[x]$ and towers $R[x]$ using segment tree, and check whether there is a tower with a lower height than tower x.

Time complexity: $O(QN \log N)$.

3.4 Subtask 4

In this subtask, $D = 1$.

Doing the greedy solution will lease towers whose both of the neighbouring towers are higher than them. We can precompute which towers have higher neighbouring towers and use prefix sum to answer the queries.

Time complexity: $O(N + Q)$

3.5 Subtask 5

In this subtask, $L = 0, R = N - 1$.

Definition 3.2. *Let $LS[x]$ and $RS[x]$ be the nearest tower to the left (and to the right) of tower x with a lower height than tower x.*

In order for tower x to be leased, there must be a tower between towers $LS[x]$ and x (also between towers x and $RS[x]$) with a height of at least $H[x] + \delta$. Therefore, the value of δ must be at most $min(h_1, h_2) - H[x]$, where h_1 is the maximum height of towers between towers $LS[x]$ and x, and h_2 is the maximum height of towers between towers x and $RS[x]$.

To answer the queries, we can use binary search to count how many towers x such that tower x is leased when the value of δ is at most D.

Time complexity: $O((N + Q) \log N)$

3.6 Subtask 6

In this subtask, the value of D is constant among all queries.

Let us solve the task for a fixed value of D. Let $A_0, A_1, \ldots$ be the index of the towers leased in the greedy solution for $L = 0, R = N - 1$ question.

For $L = l, R = r$ question, let i, j be the values such that $A_{i-1} < l \leq A_i \leq A_{i+1} \leq \ldots \leq A_j \leq r < A_{j+1}$. The towers leased in the greedy solution for this question are towers $A_i, A_{i+1}, \ldots, A_j$ and possibly two other towers: one between towers l and A_i, and another one between towers A_j and r.

Let us focus on whether we can find a tower x between towers l and A_i that we can lease. We need tower x to be able to communicate with tower A_i, so there must be an intermediary tower between them. This means $R[x]$ must be less than A_i and $L[A_i]$ must be more than x. To check whether there exists such x, we check whether $(\min_{l \leq k \leq L[A_i]-1} R[k]) < A_i$. We can get the left-hand value using segment tree.

Finding a tower between towers A_j and r can be done similarly.

Time complexity: $O((N + Q) \log N)$

3.7 Subtask 7

Let us solve this task for different possible values of D.

For a given value of D, we can reuse the solution for subtask 5 to get the index of leased towers $A_0, A_1, \ldots$ for $L = 0, R = N - 1$ question. Similar to subtask 6, for $L = l, R = r$ question, let i, j be the values such

that $A_{i-1} < l \leq A_i \leq A_{i+1} \leq \ldots \leq A_j \leq r < A_{j+1}$. The values of A_i, A_j, and $j - i$ can be computed using a 2D data structure or persistent segment tree.

To find tower x (an additional tower between towers l and A_i that we can lease), we want to check whether there exists a tower k ($x < k < A_i$) such that $H[k] > \max(H[x], H[A_i]) + D$. We can do this by having another segment tree that computes: in a node that covers the range $[l', r']$, the segment tree computes $\max_{l' \leq i \leq j \leq r'} H[j] - H[i]$. If the value of the node that covers $[l, L[A_i]]$ is at least D, then there exists x, y where $l \leq x \leq y \leq L[A_i]$ and $H[y] - H[x] \geq D$. Therefore, we can lease tower x and tower y can be used as an intermediary between tower x and tower A_i.

Finding a tower between towers A_j and r can be done similarly.

Time complexity: $O((N + Q) \log N)$.