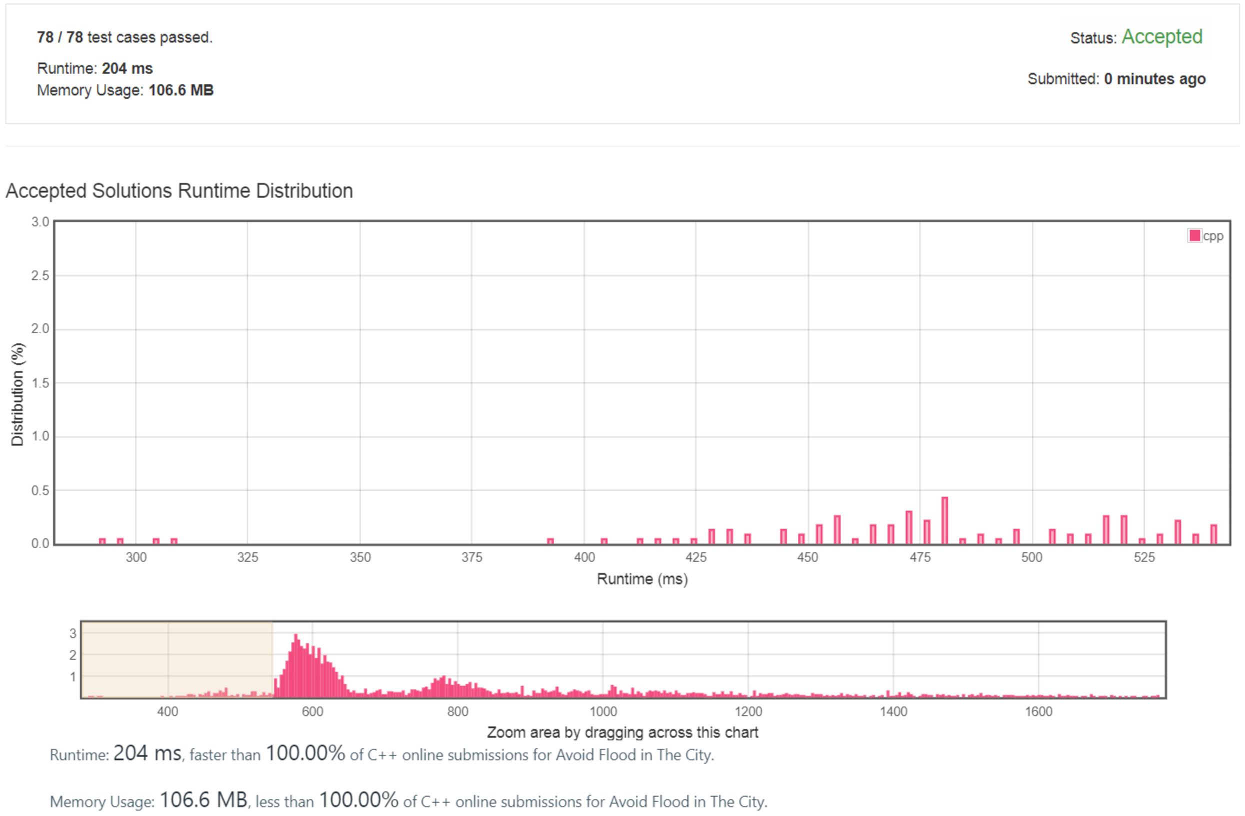
1. greedy. scan from left to right, whenever we find a pair of raining over the same lake at a previous day i and the current day j, find the earliest possible date between i and j to empty the lake, and mark that day as being used. use union-find data structure. $O(n\alpha(n))$.
2. similar to the idea in [1], by combining union-find and bit packing, the running time can be further improved to $O(n)$. we divide the array into blocks of size $w = \Omega(\log n)$, and for each block pick a representative to include in the union-find data structure. if the answer of find(i) is within the same block as i, we can use bit operation to solve in $O(1)$ time. otherwise we use the union-find data structure to query at the representative. let $n_0 = \frac{n}{w}$ and number of queries $m = O(n)$, the total running time is $O(m\alpha(m+n_0, n_0)+n) = O(n)$.



78 / 78 test cases passed.

Runtime: **204 ms**

Memory Usage: **106.6 MB**

Status: Accepted

Submitted: **0 minutes ago**

Accepted Solutions Runtime Distribution

Runtime: 204 ms, faster than 100.00% of C++ online submissions for Avoid Flood in The City.

Memory Usage: 106.6 MB, less than 100.00% of C++ online submissions for Avoid Flood in The City.

References

[1] Harold N Gabow and Robert Endre Tarjan. A linear-time algorithm for a special case of disjoint set union. *Journal of computer and system sciences*, 30(2):209–221, 1985.