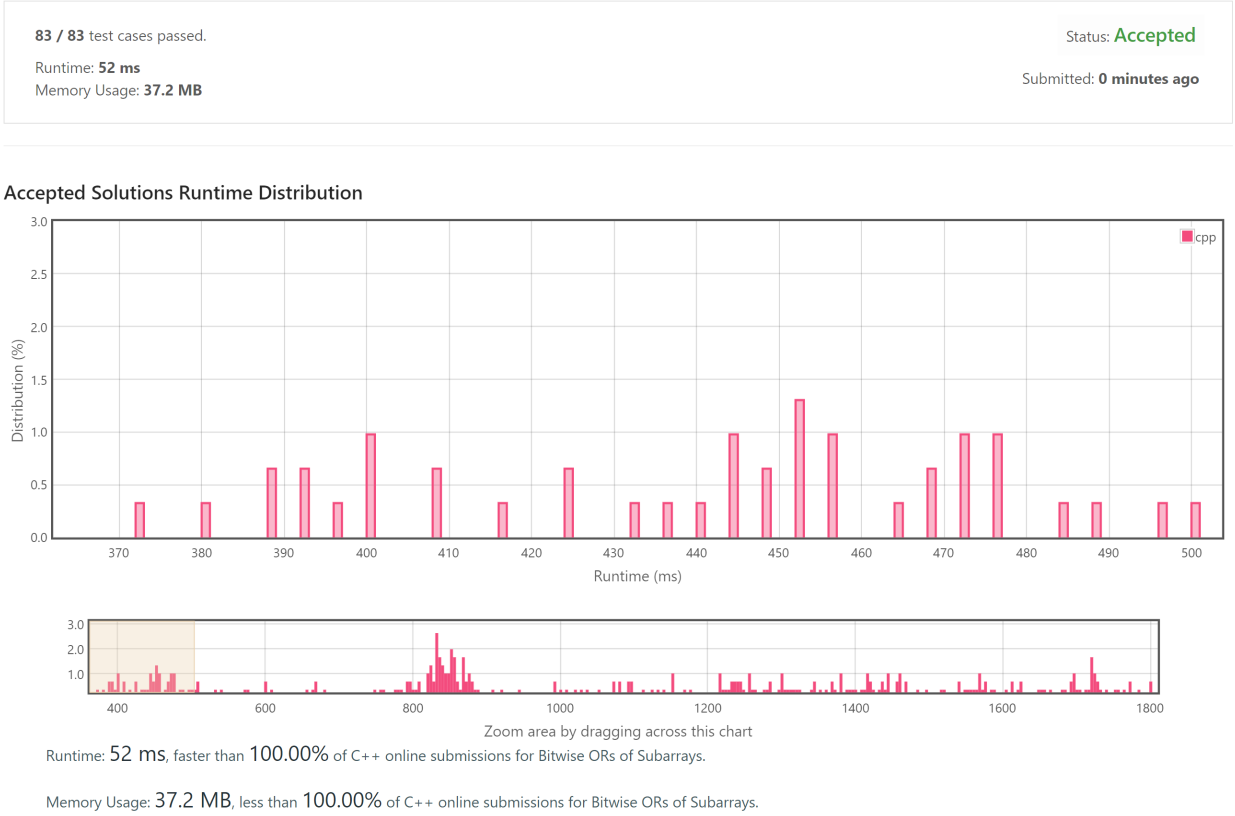
Fix the left endpoint l, if we expand the right endpoint r, the result will only change at most w times, each time change at least one bit from 0 to 1.
Let $c[l][r]$ denote the bitwise OR of $A[l]...A[r]$. Fix l, there are only $O(w)$ different $c[l][r]$'s. Also $c[l-1][r] = c[l][r]|A[l-1]$. Use $O(w)$ to transit from $c[l][*]$ to $c[l-1][*]$. Finally remove the duplicates by hashing. $O(nw)$.

lower bound: construct an example with $\Theta(nw)$ distinct results? Then it's the lower bound for all algorithms that need to remove the duplicates by enumerating them.



83 / 83 test cases passed.

Status: Accepted

Runtime: **52 ms**

Memory Usage: **37.2 MB**

Submitted: **0 minutes ago**

Accepted Solutions Runtime Distribution

Runtime: 52 ms, faster than 100.00% of C++ online submissions for Bitwise ORs of Subarrays.

Memory Usage: 37.2 MB, less than 100.00% of C++ online submissions for Bitwise ORs of Subarrays.

References