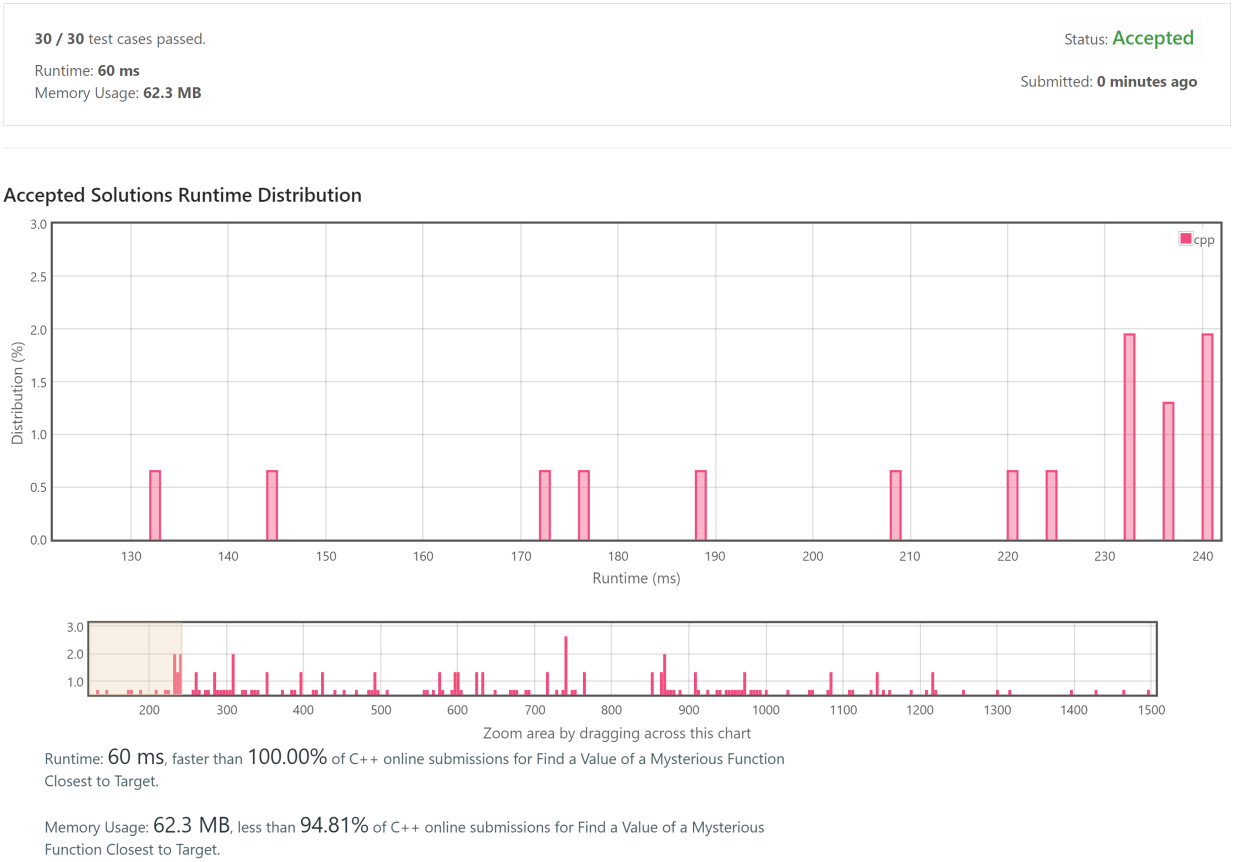
1. Fix the right endpoint and vary the left endpoint, there are only $O(w)$ different & sums, because when we move the left endpoint to the left, whenever the i-th bit becomes 0, it cannot become 1 again. $O(nw)$.
2. Fix the right endpoint and move the left endpoint to the left, the & sum is monotone decreasing. Use segment tree to find the leftmost left endpoint with & sum $\geq t$. $O(n \log n)$.
3. Maintain the smallest number $\geq t$, by using two pointers to maintain a sliding window. To maintain the current & sum, we can use two stacks, whenever the first stack is empty, move half of the elements from the second stack to the first one. Using amortized analysis, the total running time is $O(n)$.

Remark. static interval & query can be solved in (near?)$O(n)$-$O(1)$ time using standard tricks (group into blocks of size $O(\log n)$, recursively divide into miniblocks of size $O(\log \log n)$, and use bit packing within the miniblocks).



References