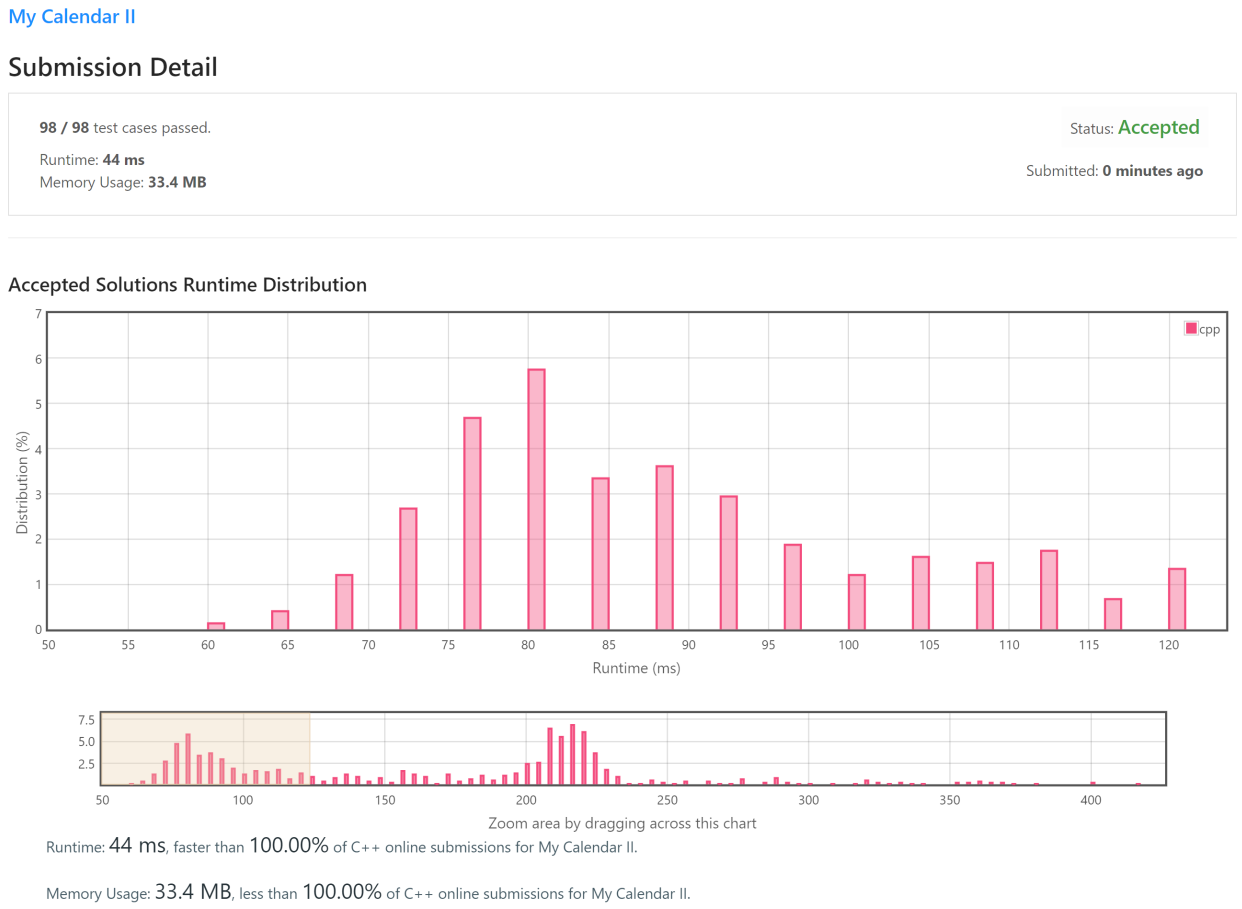
Let $k = 3$ denote the number of booking limit.
1. Balanced search tree. $O(n \log n)$.
2. Maintain the depth of each interval, using predecessor search in $O(\text{pred}(n)+\Delta)$ time per operation, where we increase the depth of Δ intervals. Also maintain a predecessor search data structure for all intervals with depth $k-1$. The total depth of the intervals is $O(n \cdot k)$, so by amortized analysis, the total running time is $O(nk + n \cdot \text{pred}(n))$.



My Calendar II

Submission Detail

98 / 98 test cases passed.

Status: Accepted

Runtime: **44 ms**

Memory Usage: **33.4 MB**

Submitted: **0 minutes ago**

Accepted Solutions Runtime Distribution

Runtime: 44 ms, faster than 100.00% of C++ online submissions for My Calendar II.

Memory Usage: 33.4 MB, less than 100.00% of C++ online submissions for My Calendar II.

References