

## C-10.11

In this case we can store candidate ID's in a balanced search tree, such as an AVL tree or red-black tree, where in addition to each ID we store in this tree the number of votes that ID has received. Initially, all such counts are 0. Then, we traverse the sequence of votes, incrementing the count for the appropriate ID with each vote. Since this data structure stored  $k$  elements, each such search and update takes  $O(\log k)$  time. Thus, the total time is  $O(n \log k)$ .