

divide and conquer, compare the middle elements of two arrays, and recurse.  $O(\log(n + m))$ .

in general, finding the  $t$ -th largest element in the union of  $k$  sorted array with respective sizes  $n_1, \dots, n_k$  takes time:

1.  $O(\sum_{i=1}^k \log n_i)$  [1].

<https://cstheory.stackexchange.com/questions/20944/select-in-union-of-sorted-arrays-already-known/20955#20955>.

2. let  $p = \min\{k, t\}$ , the running time is  $\Theta(k + p \log \frac{t}{p})$  [2].

i.e. if  $t \geq k$ ,  $O(k \log \frac{t}{k})$ . if  $t < k$ ,  $O(k)$ .

## References

- [1] Greg N Frederickson and Donald B Johnson. Generalized selection and ranking: sorted matrices. *SIAM Journal on computing*, 13(1):14–30, 1984.
- [2] Andranik Mirzaian and Eshrat Arjomandi. Selection in  $x+y$  and matrices with sorted rows and columns. *Information processing letters*, 20(1):13–17, 1985.