

## Cs301-a2

Oğuzhan Özdemir (Student)

### Problem 1 (Order statistics)

a) If we need to go with sorting and returning  $k$  smallest number the best asymptotic worst-case running time will be  $O(n \lg n)$  due to comparison sorting approach. It basically compares elements of list with each other increasing order and we can print first  $k$  smallest number with  $O(k)$  search. Total will be  $O(k + n \lg n)$

b) Using order-statistic algorithm takes  $O(n)$  time. Then using QuickSort partition takes  $O(n)$  time too. Sorting takes  $O(k \lg k)$  so total will be  $O(n + k \lg k)$ .

The selection of algorithm will be using order statistic version because  $n$  will be higher than  $k$  in all cases.

### Problem 2 (Linear-time sorting)

a) To use Radix sort on strings we need to first find longest length string then add the difference to shorter strings end. We can use 0 as dummy to extend shorter words. After reaching same length words counting sort starts from actual elements of words.

b)

5 th position: ['MERT', 'AYSU', 'SELIN', 'ERDEM', 'DILARA']

4 th position: ['MERT', 'AYSU', 'ERDEM', 'SELIN', 'DILARA']

3 th position: ['DILARA', 'ERDEM', 'SELIN', 'MERT', 'AYSU']

2 th position: ['ERDEM', 'DILARA', 'SELIN', 'MERT', 'AYSU']

1 th position: ['SELIN', 'MERT', 'DILARA', 'ERDEM', 'AYSU']

0 th and final: ['AYSU', 'DILARA', 'ERDEM', 'MERT', 'SELIN']

c) Radix sort takes longest length \* number of strings plus finding longest length of strings.