ID: hh2537; Name: Haoze He

Q1:

When , the quick sort stops. It takesFor each subarray with size k, use insertion sort, the time complexity is for each one. Sum them together, it is

To improve the result, using master theorem, we choose smaller k to make sure is not dominant comparing with . To make sure , K should be

To proof that, we select for quick sort and for insertion sort.

Thus, k =

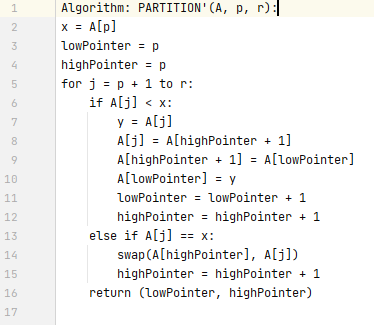
Q2.

a)

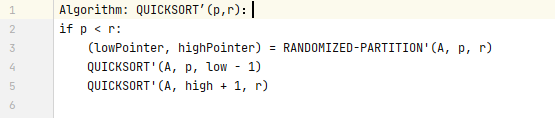
If all elements are the same, our recurrence becomes: , the time complexity becomes

b)

The pseudocode of the modified algorithm shows below:



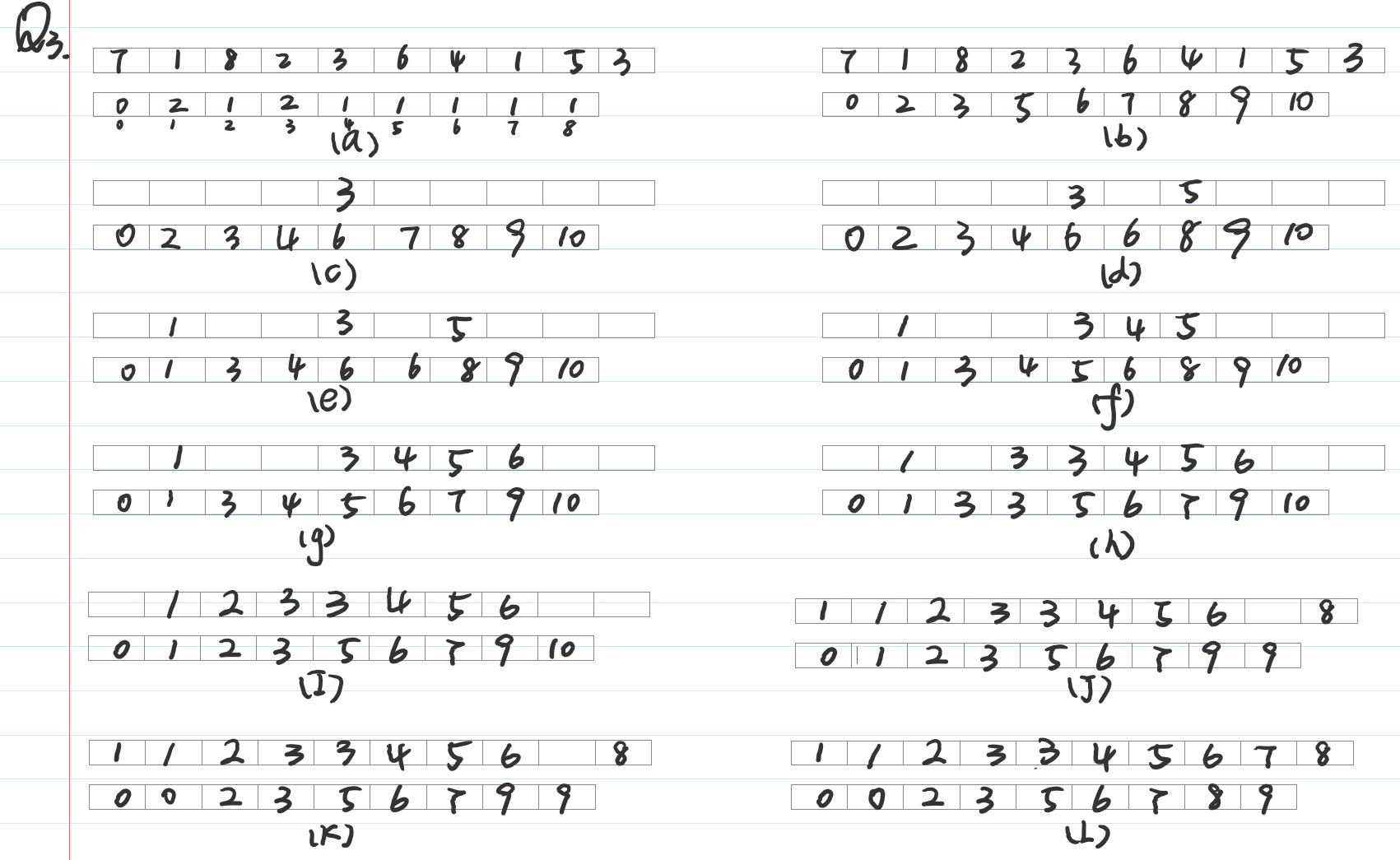
c)



d)

When all input elements are distinct, he sub-problem of QUICKSORT’ is actually not larger than the sub-problem of QUICKSORT. It sometimes performs better since it’s more balanced.

Q3.



Q4.

There are two cases, n is odd or even.

Case1, n is odd:

Case2, n is even:

Hence, proof is done.

Q5.

Sorting number running time = , running time of listing largest = .

Hence, total running time =

b)

Running time of building a max-priority queue = , running time of each time call EXTRACT-MAX = .

Hence, total running time =

c)

Running time of finding largest number and partitioning is , running time of sorting the largest number is .

Hence, the total running time is