EL9343 Homework 05

Spring 2022

Name: Sagar Patel
NETID: sp5894

Question 01

Exercise 7.4-5 in CLRS Textbook.

Solution

If we are only doing quick-sort until the problem size becomes $\leq k$, then, we will have to take $lg(\frac{n}{k})$ steps, since, as in the original analysis of randomized quick sort, we expect there to be $lg(n)$ levels to the recursion tree. Since we then just call quicksort on the entire array, we know that each element is within k of its final position. This means that an insertion sort will take the shifting of at most k elements for every element that needed to change position. This gets us the running time described. In theory, we should pick k to minimize this expression, that is, taking a derivative with respect to k, we want it to be evaluating to zero. So, $n - \frac{n}{k} = 0$. The constant of proportionality will depend on the relative size of the constants in the nk term and in the $n.lg(\frac{n}{k})$ term. In practice, we would try it with a large number of input sizes for various values of k, because there are gritty properties of the machine not considered here such as cache line size.

Question 02

Exercise 7-2 in CLRS Textbook.

2.a

Solution

Since all elements are the same, the initial random choice of index and swap change nothing. Thus, randomized quicksort's running time will be the same as that of quicksort. Since all elements are equal, $PARTITION(A, P, r)$ will always return $r - 1$. This is worst-case partitioning, so the runtime is $\Theta(n^2)$.

2.b

Solution

```
PARTITION'(A, P, r):
    x = A[p]
    i = p-1
    k = p
    for j in range (p+1,r):
        if (A[j] < x):
            i++
            k++
            SWAP A[i] and A[j]
            SWAP A[k] and A[j]

        if (A[j] == x):
            k++
            SWAP A[k] and A[j]

    return (i+1, k)
```

2.c

Solution

```
RANDOMIZED-PARTITION'
    i = RANDOM(p, r)
    SWAP A[r] and A[i]
    return PARTITION'(A,p,r)

QUICKSORT'(A,p,r):
    if (p < r):
        (q,t) = RANDOMIZED - PARTITION'
        QUICKSORT'(A,p,q-1)
        QUICKSORT'(A,t+1,r)
```

2.d

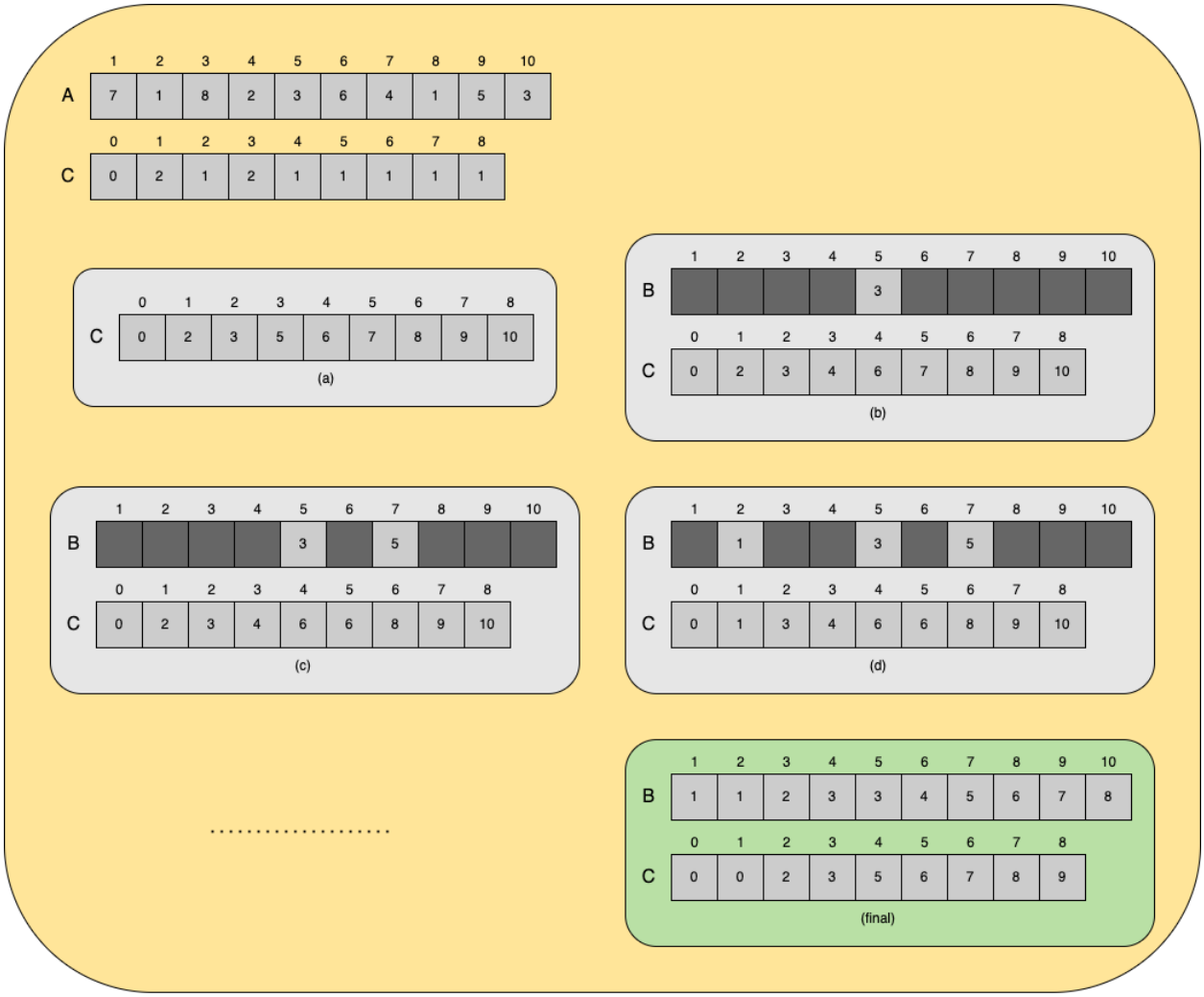
Solution

Let d be the number of distinct elements in A. The running time is dominated by the time spent in the $PARTITION$ procedure, and there can be at most d calls to $PARTITION$. If X is the number of comparisons performed in line 4 of $PARTITION$ over the entire execution of $QUICKSORT'$, then the running time is $O(d + X)$. It remains true that each pair of elements is compared at most once. If z_i is the i-th smallest element, we need to compute the probability that z_i is compared to z_j . This time, once a pivot x is chosen with $z_i \leq x \leq z_j$, we know that z_i and z_j cannot be compared at any subsequent time. This is where the analysis differs, because there could be many elements of the array equal to z_i or z_j , so the probability that z_i and z_j are compared decreases. However, the expected percentage of distinct elements in a random array tends to $1 - \frac{1}{e}$, so asymptotically the expected number of comparisons is the same.

Question 03

Similar to Figure 8.2, illustrate the operation of COUNTING-SORT on
$A = [7, 1, 8, 2, 3, 6, 4, 1, 5, 3]$

Solution



Question 04

Exercise 9.1-2 in CLRS Textbook.

Solution

If n is odd, there are –
$1 + \frac{3(n-3)}{2} + 2 = \frac{3n}{2} - \frac{3}{2}$
$\Rightarrow (\frac{3n}{2} - \frac{1}{2}) - \frac{3}{2} \Rightarrow (\frac{3n}{2}) - 2$ comparisons.

If n is even, there are –
$1 + \frac{3(n-2)}{2} = \frac{3n}{2} - 2$
$\Rightarrow (\frac{3n}{2}) - 2$ comparisons.

Question 05

Problem 9-1 in CLRS Textbook.

5.a

Solution

Sorting takes time $n.\,lg(n)$, and listing them out takes time i, so the total runtime is $O(n.\,lg(n) + i)$

5.b

Solution

Heapifying takes time $n.\,lg(n)$, and each extraction can take time $lg(n)$, so, the total runtime is $O((n + i)lg(n))$

5.c

Solution

Finding and partitioning around the i-th largest takes time n. Then, sorting the subarray of length i coming from the partition takes time $i.\,lg(i)$. So, the total runtime is $O(n + ilg(i))$.