ECE608 CHAPTER 7 PROBLEMS

1) 7.1-1
2) 7.1-4
3) 7.2-2
4) 7.2-3
5) 7.4-1
6) 7.4-4
7) 7-5

ECE608 - Chapter 7 answers

(1) CLR 7.1-1

See Figure 3 (for 2nd Edition)

(a) The initial array and position of i and j indices at the beginning of the first iteration of the for loop. Because $A[j] \leq x$, i is incremented by one to p and $A[p]$ is exchanged with itself (nothing changes).

(b) At the start of the second iteration in the for loop, we show the placement of the i and j indices. This iteration is much like the first, as are all the iterations up to the last one. (c) At the start of the last iteration in the loop, i is at position $r-2$ and j is at position $r-1$. Because $A[j] \leq x$, i is incremented by one to $r-1$ and $A[r-1]$ is exchanged with itself.

(d) $i+1=r$ is returned at line 8 after $A[r]$ is exchanged with itself.

See Figure 2 (for 3rd edition). A first few iterations and the last few iterations are shown. The conventions used are same as in the book. For example, the shaded region correspond to the second partition with elements greater than pivot.

(2) CLR 7.1-4

To make Quicksort to return A in non-increasing order, we simply replace Partition with Partition-Nonincreasing in the below.

```
PARTITION-NONINCREASING(A,p,r)
x ← A[r]
i ← p − 1
for j ← p to r − 1
    do if A[j] ≥ x
          then i ← i + 1
               exchange A[i] ↔ A[j]
```

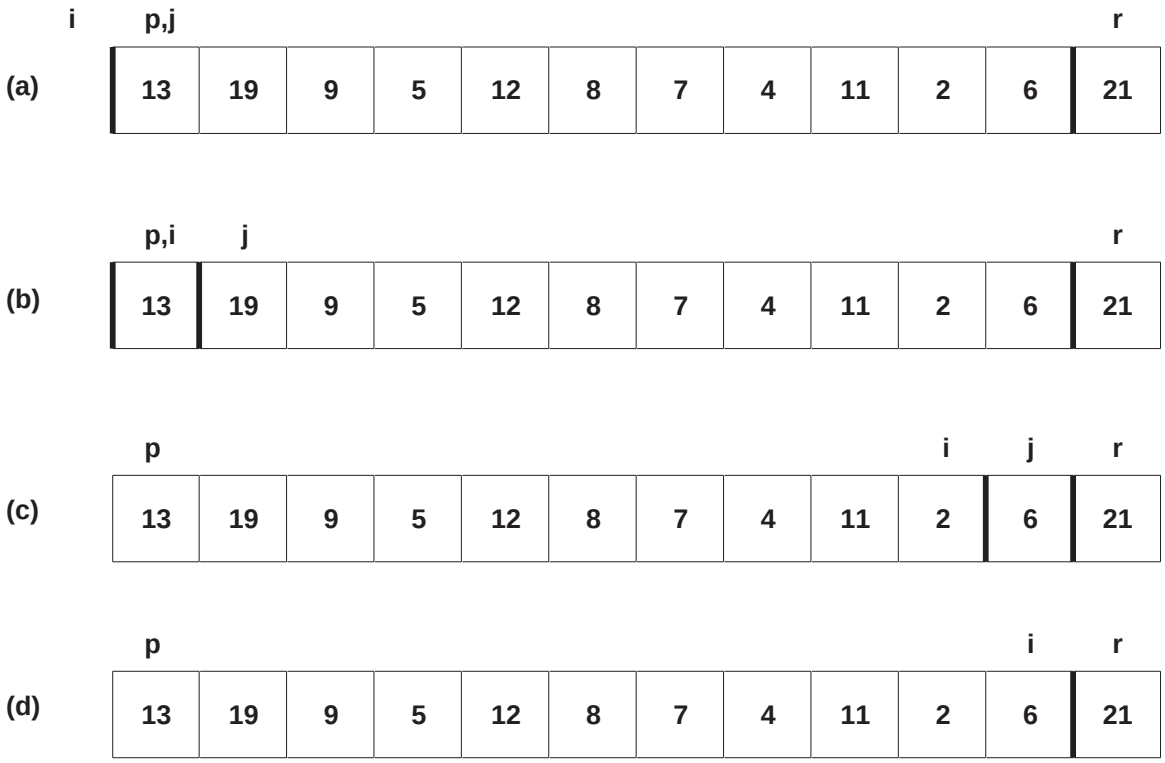

Figure 2: The operation of PARTITION on A.

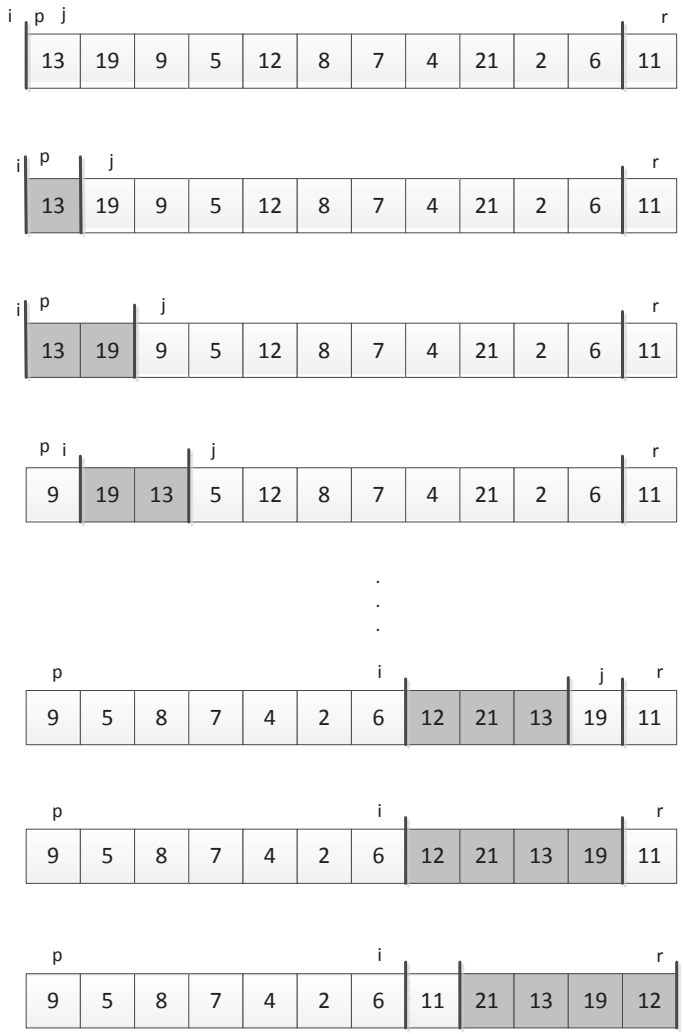

Figure 3: The operation of PARTITION on A.

```
exchange A[i + 1] ↔ A[r]
return i + 1
```

(3) CLR 7.2-2

It is a special case of an array which is sorted. According to the partition procedure, at each recursive level, the pivot is always the same as all the other elements. The first partition with values no greater than the pivot will always have $r - p$ elements, which is less than the number of total elements in the subarray by one. So the worst-case unbalanced partitioning will always happen with recursive running time $T(n) = T(n-1) + \theta(n)$. The running time is therefore $\theta(n^2)$.

(4) CLR 7.2-3

Analyze QUICKSORT when the array A is in decreasing order. The initial call to the PARTITION(A, p, r) sets the $q \leftarrow p$ which gives the worst case partition of 0 and $r - p$. On the next recursion of the QUICKSORT, the largest element is at the last cell of the subarray. PARTITION(A, p, r) sets $q = r$ which also gives a worst case partition. Now we have another subarray with decreasing order elements. Hence, QUICKSORT repeats steps above until exhausting the elements of A. The corresponding recursion is:

$\mathrm{T}(n) = \mathrm{T}(n-1) + \Theta(n) = \Theta(n^2)$

(5) CLR 7.4-1

This is very identical to the way the upper bound of the function was found in the text. Follow the same procedure and make the guess that $T(n) \geq cn^2$. Using the maxima minima method, take the first and second derivative of $q^2 + (n - q - 1)^2$. The second derivative would be 4 thus suggesting that the function is concave up for all values of q from 0 to $n - 1$. The maxima lies on the endpoints.

$T(n) \geq max(T(q) + T(n-q-1)) + \Theta(n)\ 0 \leq q \leq n-1$

$\geq max(cq^2 + c(n-q-1)^2) + \Theta(n)$

$= c(n-1)^2 + \Theta(n)$

$= cn^2 + c - 2cn + \Theta(n)$

$= cn^2 + c - 2cn + dn$

$= cn^2 + (d-2c)n + c$

$\geq cn^2$

if $2c \leq d$ and $0 \leq q \leq n-1$

$= \Omega(n^2)$

Therefore $T(n) = \Omega(n^2)$

(6) CLR 7.4-4

Since the running time of RANDOMIZED-QUICKSORT is bounded by n calls to PARTITION and X comparisons within all the calls to PARTITION, we will show that the expected running time of RANDOMIZED-QUICKSORT is $\Omega(n \lg n)$ by demonstratinging that $E[X] = \Omega(n \lg n)$.

To do the analysis, we rename the elements of A as $z_1, z_2, \ldots, z_n$ with z_i being the ith ranked element, as in the upper bound analysis of the expected running time. Our analysis then uses the indicator random variable, $X_{ij} = I\{z_i \text{ is compared to } z_j\}$ to calculate a lower bound on the number of comparisons:

$$X = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}$$

Taking the expectation of X and using linearity and Lemma 5.1, we obtain:

$$\begin{aligned} E[X] &= E[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E[X_{ij}] \\ &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} Pr\{z_i \text{ is compared to } z_j\} \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{2}{j-i+1} \\
&\geq \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{1}{j-i+1} \\
&= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{1}{j-i+1} \qquad \triangleright \text{ Let } k = j-i+1 \text{ with } k \text{ ranging from 2 to } n-i+1 \\
&= \sum_{i=1}^{n-1}(\sum_{k=1}^{n-i+1}\frac{1}{k}-1) \\
&\geq \sum_{i=1}^{n-1}(\sum_{k=1}^{n-i}\frac{1}{k}-1) \\
&= \sum_{i=1}^{n-1}(\Omega(\lg(n-i))-1) \qquad \triangleright \text{ By A.2-3} \\
&= \sum_{i=1}^{n-1}\Omega(\lg(n-i))-(n-1) \\
&\geq \sum_{i=1}^{n-1}\Omega(\lg(n-i))-n \\
&= \Omega(\sum_{i=1}^{n-1}\lg(n-i))-n \\
&= \Omega(\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor}\lg(n-i)+\sum_{i=\lfloor\frac{n}{2}\rfloor+1}^{n-1}\lg(n-i))-n \\
&\geq \Omega(\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor}\lg(n-\frac{n}{2})+\sum_{i=\lfloor\frac{n}{2}\rfloor+1}^{n-1}\lg(n-(n-1)))-n \\
&= \Omega(\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor}\lg(\frac{n}{2}))-n \\
&= \Omega(\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor}(\lg n-\lg 2))-n \\
&= \Omega(\lfloor\frac{n}{2}\rfloor\lg n-\lfloor\frac{n}{2}\rfloor)-n \\
&= \Omega(\lfloor\frac{n}{2}\rfloor(\lg n-1))-n \\
&= \Omega(n\lg n)
\end{aligned}
$$

Hence, we can conclude the expected running time is $\Omega(n \lg n)$.

(7) CLR 7-5

a. Given $A'[i]$ is the pivot x selected, the number of ways to select the other two elements is $(i-1)(n-i)$, because we can choose any one of the $i-1$ elements which are smaller than $A'[i]$ and any one of the $n-i$ elements which are larger than $A'[i]$ to ensure that $A'[i]$ is the median of the three selected elements.

The number of ways to select the three elements is $\sum_{k=2}^{n-1}(k-1)(n-k)$ or, equivalently, $\binom{n}{3}$).

So the probability of $A'[i]$ becoming the pivot x is:

$\frac{(i-1)(n-i)}{\binom{n}{3}}$

$\frac{=(i-1)(n-i)}{\frac{1}{6}(n^3-3n^2+2n)}$

b. The probability of choosing the pivot as $x = A'[\lfloor (n+1)/2 \rfloor]$ for the conventional method is $1/n$.

The probability of choosing the pivot as $x = A'[\lfloor (n+1)/2 \rfloor]$ for the median-of-3 method is $\frac{(\lfloor (n+1)/2 \rfloor - 1)(n - \lfloor (n+1)/2 \rfloor)}{\frac{1}{6}(n^3-3n^2+2n)}$.

So

$\lim_{n\to\infty} \frac{\frac{(\lfloor (n+1)/2 \rfloor - 1)(n - \lfloor (n+1)/2 \rfloor)}{\frac{1}{6}(n^3-3n^2+2n)}}{1/n} = 1.5$

This means we can increase the likelihood by 50%.

c. The probability of getting a good split by the conventional method is $\frac{1/n \cdot (2n/3 - n/3 + 1) = n+3}{3n}$.

The probability of getting a good split by the median-of-3 partition method is $\frac{\sum_{i=n/3}^{2n/3}(i-1)(n-i)}{\sum_{k=2}^{n-1}(k-1)(n-k)}$.

Approximating the numerator by integration,

$\sum_{i=n/3}^{2n/3}(i-1)(n-i)$

$\simeq \int_{n/3}^{2n/3}(k-1)(n-k)dk$

$= \frac{13}{162}n^3 + \frac{1}{6}n^2$

So $\lim_{n\to\infty} \frac{\sum_{i=n/3}^{2n/3}(i-1)(n-i)}{\sum_{k=2}^{n-1}(k-1)(n-k)} = 13/27$.

And since $\lim_{n\to\infty} \frac{n+3}{3n} = 1/3$.

The likelihood of getting a good split will increase by $\frac{13/27}{1/3} - 1$=44.4%.

d. The median-of-3 method only optimizes the choosing of pivot so that the resulting partition will have a relatively high probability of becoming balanced partition. However, even if the partition is well balanced (divided by 2), the running time will still be $\Omega(nlgn)$.