**Question 1**: Design and Analysis of the algorithms

1. **Algorithm** TwoColor(A,n)  
   Input: Array A of n balls with 2 color  
   OutPut: Array A with all the balls sorted accouring to their color.  
   n🡨A.lengh  
   **if n=0 || n=1**  
    return A  
   p🡨A[0]   
   i🡨 1  
   j🡨 n   
   **while i<=j**  
    **if A[i]!=A[j]**  
    **if p=A[j]**  
    SWAP(A[i],A[j])  
    i++  
    j--  
    j--  
   **return A**  
   🡪My Algorithm is in place. And the time complicity is O(n).
2. **Algorithm** 4ColorAnd3Color(A,n)  
   ***Input***: Array A of n balls with 3 or more Different Colors  
   ***Output***: Array A sorted according to their color  
   n🡨 A.length  
   **if n=0 || n=1**  
    return A  
   p🡨A[0]  
   i🡨 1  
   j🡨 n  
   **while i!=n**  
    **if A[i]!=A[j]**  
    **if p = A[j]**  
    SWAP(A[i],A[j])  
    i++  
     
    **if i>=j**  
    p=A[i]  
    j=n  
    i++  
    j--  
   **return A**

* ThisAlgorithm is working well for both B and C. Also it’s in place. And the time complicity is O(n).

**Question 2:**

1. **A= {1, 2, 3, 4, 5, 6, 7, 8, 9}  
   3(8)/4 = 6  
   pivot = A[6] = 7   
   L= {1,2,3,4,5,6}** size which is equal to 6   
   🡪 So we can say this is Bad Call.  
   🡪we don’t have to do the sorting its already sorted
2. **A={8, 7, 6, 5, 4, 3, 2, 1, 9}  
   let’s pick a p by using he median number= 8,9,4  
   p=4  
   {8,7,6,5,9,3,2,1,4} = A  
    {1,7,6,5,9,3,2,8,4}  
    {1,2,6,5,9,3,7,8,4}** **{1,2,3,5,9,6,7,8,4}  
    {1,2,3,4,9,6,7,8,5}**again choose a pivot using median: 5 **{9,6,5,8,7}  
    {5,6,9,8,7}  
    {5,6,7,8,9}**🡪 Then join them up = {1,2,3,4,5,6,7,8,9}

1. **{9, 1, 8, 2, 7, 3, 6, 4, 5}**

Let’s Select pivot

* Position A[0]+A[8]/2= 0+8/2=4

              =>median of **9,7,5 is 7**

**Pivot is 7**

**{9, 1, 8, 2, 7, 3, 6, 4, 5}**

**Swap(7,5)**

* **{9, 1, 8, 2, 7, 3, 6, 4, 5}**
* **{4, 1, 6, 2, 5, 3}, 7, {9, 8}**
* **{2, 1, 6, 4, 5, 3}, 7, {9, 8}**
* **{2, 1,} 3 { 4, 5, 6}, 7, {9, 8}**
* **{1,2} 3 {4,5,6},7,{8,9}**
* **So the sorted will be   {1,2,3, 4 ,5,6,7, 8 ,9}**

**D. {5, 1, 4, 2, 3, 9, 7, 6, 8}**

Let’s Select pivot

* Position A[0]+A[8]/2= 0+8/2=4

              =>median of **5,3,8 is 5**

* **{3,1,4,2,59,7,6,8}**
* **{3,1,4,2,8,9,7,6,5}**
* **{3,1,4,2}5{9,7,6,8}**
* **{1,2,4,3,5,6,7,8,9}**
* **So the sorted will be   {1,2,3, 4 ,5,6,7, 8 ,9}**

**Question 3:**

**N.B🡪 p=pivot**

1. **A=** **{1, 2, 3, 4, 5, 6, 7, 8, 9} k = 5  
   p= 3(8)/4=6 =>7**

**E= {7}  
L={1,2,3,4,5,6}** size is bigger than k so we have our number in here.  
 p=3(5)/4=3 => 4  
E= {4}  
G={5,6} again pic a number for pivot  
 p= 3(1)/4=1=>5  
 E={5}=K=5  
 🡪 we found the number 5th smallest is 5.

1. **A= {8, 7, 6, 5, 4, 3, 2, 1, 9} k = 3  
   p=3(8)/4= 6 => 2  
   E= {2}  
   L= {1}** Size is 1 so definitely our number is in the G Array. **G= {9,3,4,5,6,7,8}** p = 3(6)/4 = 4 => 6  
    E={6}  
    L={3,4,5}  
    p= 3(2)/4 = 1 => 3  
    E={3} = k = 3  
    🡪 so we got the 3th smallest number is 3.
2. **A= {9, 1, 8, 2, 7, 3, 6, 4, 5} k = 8  
   p=3(8)/4 = 6 => 6  
   E={6}  
   L={1,2,3,4,5}  
   G={7,8,9}  
    🡪**so, k > |L|+|E|=8 > 5+1, that mean our number is in G in second place.P= 3(2)/4 = 1 => 8  
   E= {8}  
    E=8=K=8  
   🡪**so, the 8th smallest number is 8.**
3. **A={5, 1, 4, 2, 3, 9, 7, 6, 8} k = 5  
   p=3(8)/4 = 6 =>7  
   E={7}  
   L={1,2,3,4,5,6}  
   G={8,9}  
    🡪** So, K < |L|+|E|= 5 < 6+1, that mean our number is in L 5th place.  
    P= 3(5)/4 = 3 => 4  
    E={4}  
    L={1,2,3}  
    G={5,6}  
    🡪so, K > |L|+|E| = 5 > 3+1, that mean our number is in G 1st place.  
    Take the first pivot:  
    E={5} E=K=5  
    🡪 so, we find the 5th smallest number which is 5.

**Question 4:**

Good Self call: the size of L and G are each less than 2n/3

Bad Self Call: one of L and G has size greater than or equal to 2/3n

Since 2/3 is greater than ½ and less than ¾  we can say it probability is good self call

Let prove this.

The height of recursion tree is one less than the length of the descending sequence

    That means n,2/3n,(2/3)2n,…..,1,0

              n+2/3n+(2/3)2n+(2/3)3n+………+1

                          n[1+2/3+(2/3)2+(2/3)3+……….+ 1]

                           n[1+]]

                           n[1/1-x]

                            n[1/1-2/3]

                            n[1/3]

                            3n

                      T(n)=O(n)

                At each level of the recursion tree, total processing time is O(n)

                Therefore, total running time in the good case is O(nlog n)

**And this 3n is less than 4n is better than 4/3n but since this is constant there is no difference b/n them.**

**And using master formula we can say also that 2/3n**

**Is O(n)**

**Let prove it**

**T(n)≤T(2/3)n+ Cnk**

**Therefore a=1**

**B= 3**

**K=1**

**Bk=3**

**That means a< bk**

**Therefore, it will be Θ(n).**