**Assignment Six**

**Question 1**

a.

**Algorithm** splitToys(A, start, stop)

Let p be the index of the pivot such that

Swap(A, p, stop)

While (i<=j) do

if (A[stop] == 0)

while (i<stop & A[i] > A[stop]) i++

while(j>=start & A[j] <= A[stop]) j- -

else

while (i<stop & A[i] < A[stop]) i++

while(j>=start & A[j] >= A[stop]) j- -

if(i < j)

swap(A, i, j)

i++

j- -

swap ( A, i, stop)

The algorithm is in place

Time Complexity :

b. Let Red = 0, Blue = 1, Green =2

1 2 1 1 0 0 2

1 2 1 2 0 0 1

swap(A[i], A[j])

0 2 1 2 0 1 1

swap(A[i], A[j])

0 0 1 2 2 1 1

0 0 1 2 2 1 1

0 0 1 2 2 1 1

0 0 1 2 2 1 1

0 0 1 2 1 1 2

swap(A[i], A[j])

0 0 1 1 1 2 2

swap(A[i], pivot)

0 0 1 1 1 2 2

Yes, the algorithm is in place.

Time Complexity:

c. Let Blue = 0, Red = 1, Green = 2, Yellow = 3

1 1 0 2 3 2 1 0 0 3 2

1 1 0 2 3 2 1 0 0 3 2

swap (A[i], A[j])

1 1 0 0 3 2 1 0 2 3 2

swap (A[i], A[j])

1 1 0 0 0 2 1 3 2 3 2

swap (A[i], A[j])

1 1 0 0 0 1 2 3 2 3 2

swap (A[i],pivot)

1 1 0 0 0 1 [2] 3 2 3 2

1 1 0 0 0 1 [2] 3 2 3 2

swap

0 1 0 0 1 1 [2] 3 2 3 2

swap(A[i], A[j])

0 0 0 1 1 1 [2] 2 2 3 3

0 0 0 1 1 1 2 2 2 3 3

After this point the steps can be repeated but it is not important as it can be seen that they are all sorted.

**Question 2**

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

1 2 3 4 **5** 6 7 8 9

1 2 3 4 9 6 7 8 **5**

1 2 3 4 9 6 7 8 5

[1] 2 3 4] [5] [6 7 8 9]

1 4 3 2 [5] 6 9 8 7

1 [2] 3 4 [5] 6 [7] 8 9

1 [2] 4 3 [5] 6 [7] 9 8

1 2 3 4 5 6 7 8 9

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

8 7 6 5 9 3 2 1 4

swap

1 7 6 5 9 3 2 8 4

swap

1 2 6 5 9 3 7 8 4

1 2 6 5 9 3 7 8 4

swap

1 2 3 5 9 6 7 8 4

1 2 3 5 9 6 7 8 4

1 2 3 5 9 6 7 8 4

swap (A[i], pivot)

1 2 3 4 9 6 7 8 5

1 3 2 [4] 9 6 5 8 7

1 3 2 [4] 9 6 5 8 7

swap swap (A[i], A[j])

1 [2] 3 [4] 5 6 9 8 7

1 [2] 3 [4] 5 6 9 8 7

swap(A[i], pivot)

[1] [2] [3] [4] 6 5 [7] 9 8

1 2 3 4 5 6 7 8 9

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

9 1 8 2 5 3 6 4 7

swap (A[i], A[j])

4 1 8 2 5 3 6 9 7

swap (A[i], A[j])

4 1 6 2 5 3 8 9 7

4 1 6 2 5 3 [7] 9 8

3 1 6 2 5 4 [7] 8 9

swap

3 1 2 6 5 4 [7] [8] [9]

swap (A[i], pivot)

3 1 2 [4] 5 6 [7] [8] [9]

3 1 2 4 5 6 [7] [8] [9]

swap(A[i], A(j])

1 3 2 4 5 6 [7] [8] [9]

swap(A[i], A[j])

1 2 3 4 5 6 7 8 9

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

8 1 4 2 3 9 7 6 5

8 1 4 2 3 9 7 6 5

swap (A[i], A[j])

3 1 4 2 8 9 7 6 5

3 1 4 2 8 9 7 6 5

swap (A[i], A[j])

3 1 4 2 [5] 9 7 6 8

swap (A[i], A[j])

3 1 4 2 [5] 6 7 9 8

swap

1 3 4 2 [5] 6 7 9 8

swap swap(A[i], A[j])

1 2 4 3 [5] 6 7 8 9

1 2 3 4 [5] 7 6 8 9

swap(A[i], A[j])

1 2 3 4 [5] 6 7 8 9

**Question 3**

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

1 2 3 4 9 6 7 8 5

1 2 3 4 9 6 7 8 5

1 2 3 4 [5] 6 7 8 9

The 5th element is 5.

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

8 7 6 5 4 3 2 1 9

9 7 6 5 4 3 2 1 8

1 7 6 5 4 3 2 9 8

1 7 6 5 4 3 2 8 9

1 7 6 5 4 3 2 8 9

1 7 6 5 4 3 2 8 9

1 2 6 5 4 3 7 8 9

1 2 7 5 4 3 6 8 9

1 2 3 5 4 7 6 8 9

1 2 3 5 4 7 6 8 9

1 2 3 5 4 6 7 8 9

1 2 [3] 4 5 6 7 8 9

The 3rd element is 3.

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

9 1 8 2 7 3 6 4 5

9 1 8 2 5 3 6 4 7

swap(A[i], A[j])

4 1 8 2 5 3 6 9 7

swap(A[i], A[j])

4 1 6 2 5 3 8 9 7

swap(A[i], A[j])

4 1 6 2 5 3 7 9 8

4 1 6 2 5 3 7 [8] 9

The 8th element is 8.

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

8 1 4 2 3 9 7 6 5

3 1 4 2 8 9 7 6 5

3 1 4 2 8 9 7 6 5

3 1 4 2 [5] 9 7 5 8

[k=1, k=4] k=5 [k=6 up to k=9]

The 5th element is 5.

**Question 4**

This length is

At each level of the recursion tree, total processing time is . Therefore, the total running time is the good case is