**Lab W2D1**

**Question 1**

a) Algorithm toySort(A, start, stop, colorSet)

              If ( colorSet.size() == 1) return

             LeftColorSet <- firstHalfoFColorSet

              RightColorSet <- SecondHalfoFColorSet

              i <- start

              j <- stop

              while (i <= j) do

                             while (i < stop & A[i].color is in LeftColorSet ) i++

                             while (j > start & A[j].color is in RightColorSet) j--

                             If (i  < j)

                                            swap(A, i++, j--)

              toySort(A, start, i - 1, LeftColorSet)

              toySort(A, i, stop, RightColorSet)

Time complexity is O(n log k)

Space Complexity is O(1)

b) Algorithm toySort(A, start, stop, colorSet)

              If ( colorSet.size() == 1) return

             LeftColorSet <- firstHalfoFColorSet

              RightColorSet <- SecondHalfoFColorSet

              i <- start

              j <- stop

              while (i <= j) do

                             while (i < stop & A[i].color is in LeftColorSet ) i++

                             while (j > start & A[j].color is in RightColorSet) j--

                             If (i  < j)

                                            swap(A, i++, j--)

              toySort(A, start, i - 1, LeftColorSet)

              toySort(A, i, stop, RightColorSet)

Time complexity is O(n log k)

Space Complexity is O(1)

c) Algorithm toySort(A, start, stop, colorSet)

              If ( colorSet.size() == 1) return

             LeftColorSet <- firstHalfoFColorSet

              RightColorSet <- SecondHalfoFColorSet

              i <- start

              j <- stop

              while (i <= j) do

                             while (i < stop & A[i].color is in LeftColorSet ) i++

                             while (j > start & A[j].color is in RightColorSet) j--

                             If (i  < j)

                                            swap(A, i++, j--)

              toySort(A, start, i - 1, LeftColorSet)

              toySort(A, i, stop, RightColorSet)

Time complexity is O(n log k)

Space Complexity is O(1)

**Question 2**

(a){1, 2, 3, 4, 5, 6, 7, 8, 9} pivot - 5

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

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

𝑖→ 𝑖→ 𝑖→ 𝑖 𝑗⟵𝑗 ⟵𝑗⟵𝑗

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

𝑖→ 𝑖 𝑖→ 𝑖

𝑗 ⟵𝑗 ⟵𝑗 𝑗 ⟵𝑗 ⟵𝑗 (Swap)

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

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

𝑖,𝑗−𝑠𝑤𝑎𝑝 𝑖,𝑗−𝑠𝑤𝑎𝑝

1 2 3 4 5 6 7 8 9

**(b) {8, 7, 6, 5, 4, 3, 2, 1, 9} Pivot = 4**

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

𝑖→ ⟵𝑗 swap

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

𝑖 𝑗 swap

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

𝑖→ ⟵𝑗

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

𝑖 j swap

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

𝑖 𝑗

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

𝑖,𝑗

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

𝑗 𝑖 swap (A[i], pivot)

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

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

𝑖 𝑗 𝑖 ⟵𝑗

1 3 2 [4] 9 6 5 8 7

𝑗 𝑖 swap 𝑖 𝑗 swap (A[i], A[j])

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

𝑖,𝑗

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

𝑗 𝑖 swap(A[i], pivot)

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

𝑖,𝑗 𝑖,𝑗

1 2 3 4 5 6 7 8 9

C {9, 1, 8, 2, 7, 3, 6, 4, 5} **Pivot = 7**

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

𝑖→ ⟵𝑗 swap (A[i], A[j])

4 1 8 2 5 3 6 9 7

𝑖→ 𝑖→ ⟵𝑗 swap (A[i], A[j])

4 1 6 2 5 3 8 9 7

𝑖→ 𝑖→ 𝑖,𝑗 →𝑖

4 1 6 2 5 3 [7] 9 8

3 1 6 2 5 4 [7] 8 9

𝑖→ 𝑖→ 𝑖→ ⟵𝑗 ⟵𝑗 swap 𝑖,𝑗 →𝑖

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

𝑗 𝑖 swap (A[i], pivot)

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

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

𝑖→ ⟵𝑗 swap(A[i], A(j])

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

𝑗 𝑖 swap(A[i], A[j])

1 2 3 4 5 6 7 8 9

(d) {5, 1, 4, 2, 3, 9, 7, 6, 8}

8 1 4 2 3 9 7 6 5

𝑖→ ⟵𝑗

8 1 4 2 3 9 7 6 5

𝑖 𝑗 ⟵𝑗 ⟵𝑗 swap (A[i], A[j])

3 1 4 2 8 9 7 6 5

𝑖→ 𝑗

3 1 4 2 8 9 7 6 5

𝑖 𝑗 swap (A[i], A[j])

3 1 4 2 [5] 9 7 6 8

𝑖 ⟵𝑗 𝑖→ ⟵𝑗 swap (A[i], A[j])

3 1 4 2 [5] 6 7 9 8

𝑖→ ⟵𝑗 swap 𝑖,𝑗

1 3 4 2 [5] 6 7 9 8

𝑗,𝑖 swap 𝑗,𝑖 swap(A[i], A[j])

1 2 4 3 [5] 6 7 8 9

1 2 3 4 [5] 7 6 8 9

𝑖,𝑗 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**

A[0] = 1, A[8] =9 , mid = A[0+8/2] = A[4] =5 , the median of three values { 1, 9,5} = 5;

So, Pivot is **5**.

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

i → i → i→ ji→ i j ⟵j ⟵ ⟵j

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

|L| = 4, |E| = 5, |G| = 4

Since |L| + |E| = k, the 5th element is 5.

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

A[0] =8 , A[8]= 9 , mid = A[0+8/2] = A[4] =4 and the median of three values { 8,9,4} = 8;

So, Pivot is **5**.

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

i → j swap(i,j)

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

i → i → i → i → i → i→ ji→ ij

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

|L| = 7, |E| = 8, |G| = 1. Since k < 7, recursively call QuickSelect(L,3)

1 7 6 5 4 3 2

A[0] = 1, A[6] =2, A[0+6/2] = A[3] =5 and the median of three values {1,5,2} = 2 , So Pivot is 2.

1 7 6 5 4 3  **2**

ji→ ij⟵ j ⟵ j⟵ j ⟵ j

1 [2] 6 5 4 3 **7**

|L| = 1, |E| = 1, |G| = 5. Since k > |L| +|E|, recursively call QuickSelect (G,1)

6 5 4 3 **7**

A[0] = 6, A[4] =7,mid=A[2] = 4 and the median of three values {6,4,4} = 4 , So Pivot is 4.

6 5 7 3 **4**

i j swap(I,j)

3 5 7 6 **4**

i j→ ji j

3 [4] 7 6 **5**

the 3rd element is 3.

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

A[0] =9, A[8] = 5, mid = A[4] =7 and the median of three values {9,5,7} = 7 , So Pivot is 7.

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

I j

4 1 8 2 5 3 6 9 7

i → i→ i j ⟵ j

4 1 6 2 5 3 8 9 7

i → i→ i→ i→ i→ ij ⟵ ji ⟵ j

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

|L| = 6, |E| = 1, |G| = 2. Since k > |L| +|E|, recursively call QuickSelect (G,1)

9 8

A[0] = 9, A[1]=8 , A[0]=9 and the median of three values {9,8,9} = 9 , So Pivot is 9.

[8] [9]

|L| = 1. |E| = 1. |G| = 0.

Since |L|+|E| = k = 2 and the 8th element is 8.

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

A[0] =5, A[8] = 8, mid = A[4] =3 and the median of three values {5,8,3} = 5, So Pivot is 5.

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

i j ⟵j ⟵ j ⟵ j swap(I,j)

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

I→ i→ i j→ i

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

|L| = 4, |E| = 1, |G| = 4. Since k = |L| +|E|, the 5th element is 5.