**Course: Algorithm**
**Prof. Prem Nair**
**Student: Binh Van Tran**
**ID: 986648**
**Homework: Lab 6**

In place algorithm: The space complexity of the extra space is O(1)

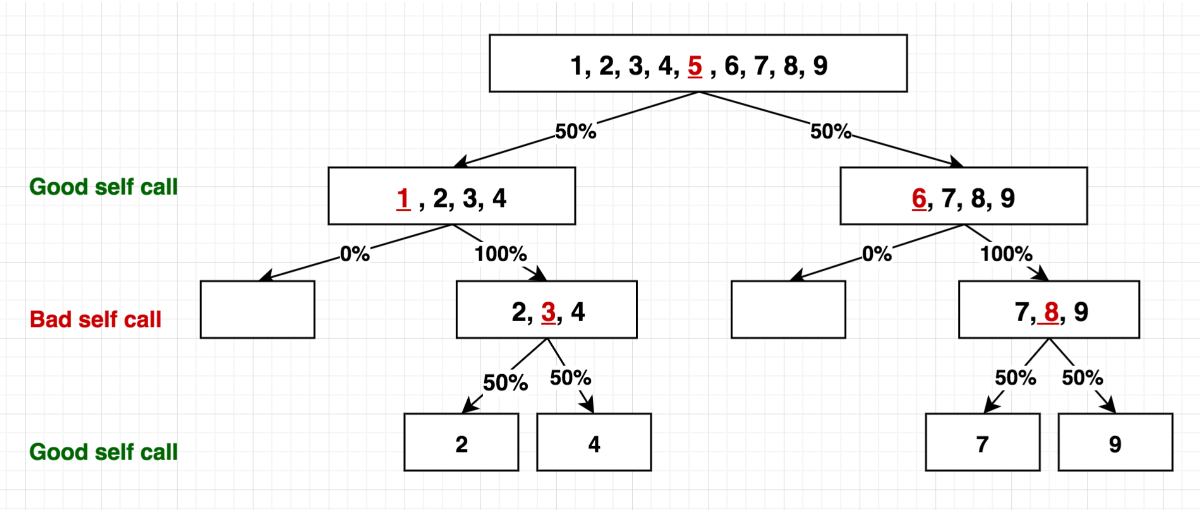
1. **Question 1 –** Design and Analysis of the algorithms
   *a. "Wooden blocks toys"*
   Say, Blue block is 1, Red block is 0

**deviseBlocks**$(blocks, n)$
$\boldsymbol{Input}$ $array$ of n $0$ $and$ $1$
$\boldsymbol{Output}$ $array$ $of$ 0 at a side, 1 at the other side
$i \leftarrow 0$
$j \leftarrow n-1$
$\boldsymbol{while}$ $i < j$ $\boldsymbol{do}$
    $\boldsymbol{while}$ $A[i] == 0$ **do** $i \leftarrow i+1$
    $\boldsymbol{while}$ $A[j] == 1$ **do** $j \leftarrow j-1$
    $swap(A[i], A[j])$
    $i \leftarrow i+1$
    $j \leftarrow j-1$
$\boldsymbol{return}$ $A$

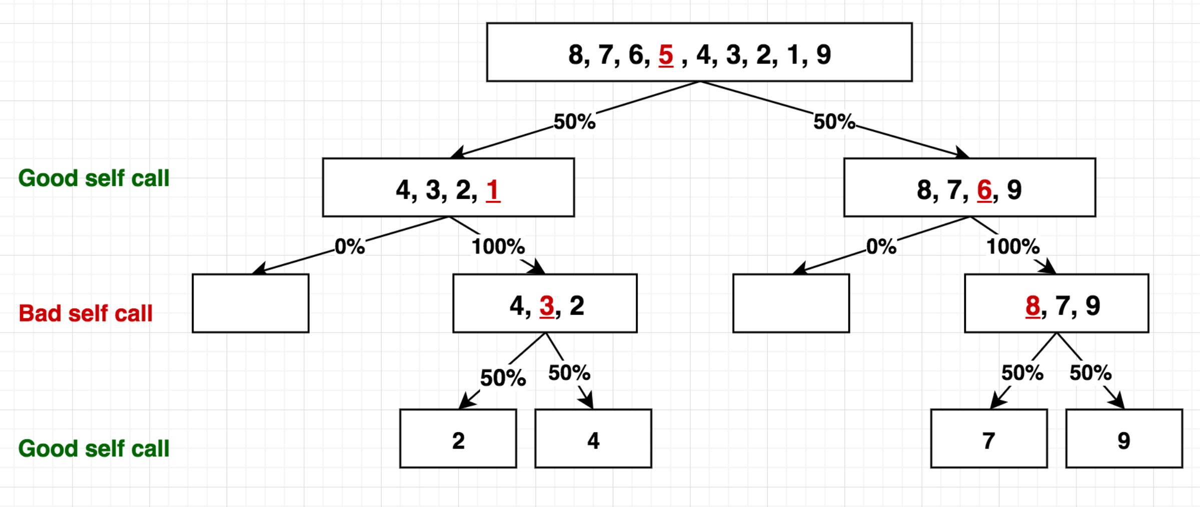
This algorithm does not need any additional space, **means in-place algorithm**
Time complexity is *O(n)*

   b. *Solve the problem for three different colors*: *Blue, Red* and *Green*
   Say Blue is 1, Red is 2, Green is 3. The problem now is sorting problem with the blocks (Blue, Red, Green) is the array of 1, 2 or 3
   We can simply use Quick Sort algorithm to divide the blocks into 3 parts. As we all know, Quick Sort algorithm can be done in-place with time complexity is *O(nlogn)*

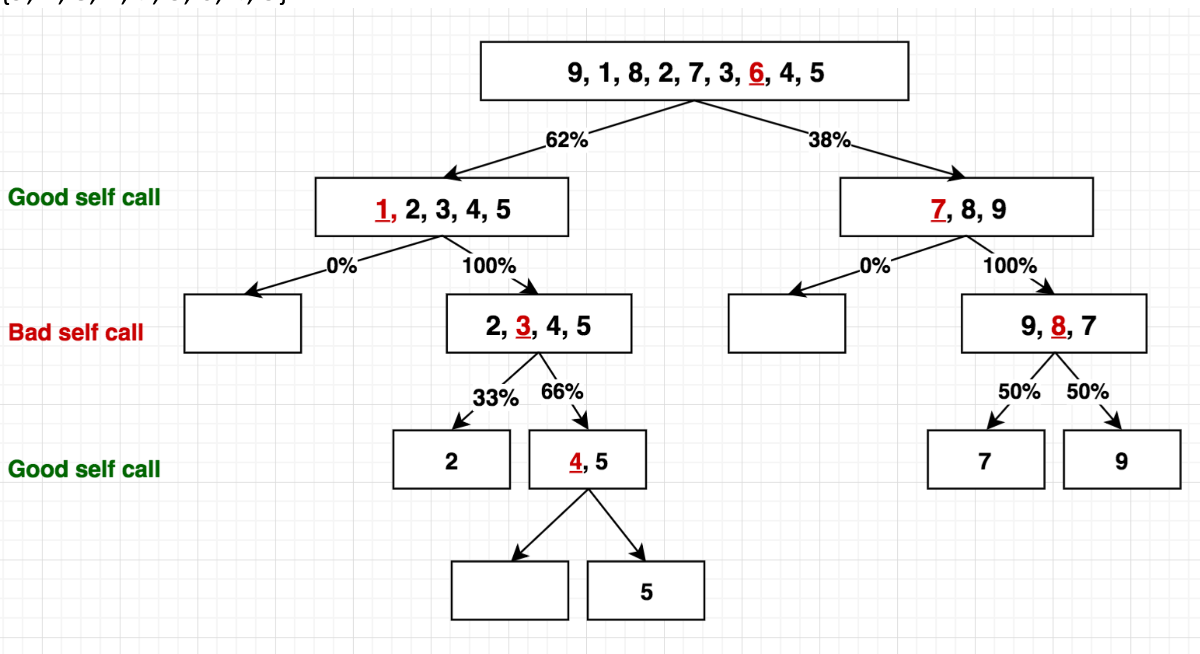
   c. *Solve the problem for three different colors*: *Blue, Red, Green* and *Yellow*
   This problem is as same as the *(b)* so we can also use Quick Sort algorithm to solve this problem. We can do it in-place with time complexity of *O(nlogn)*
2. **Question 2 -** Illustrate Quick sort, pick a pivot so that they lead to alternating between "Good Self Call" and "Bad Self Call"
   a. {1, 2, 3, 4, 5, 6, 7, 8, 9}



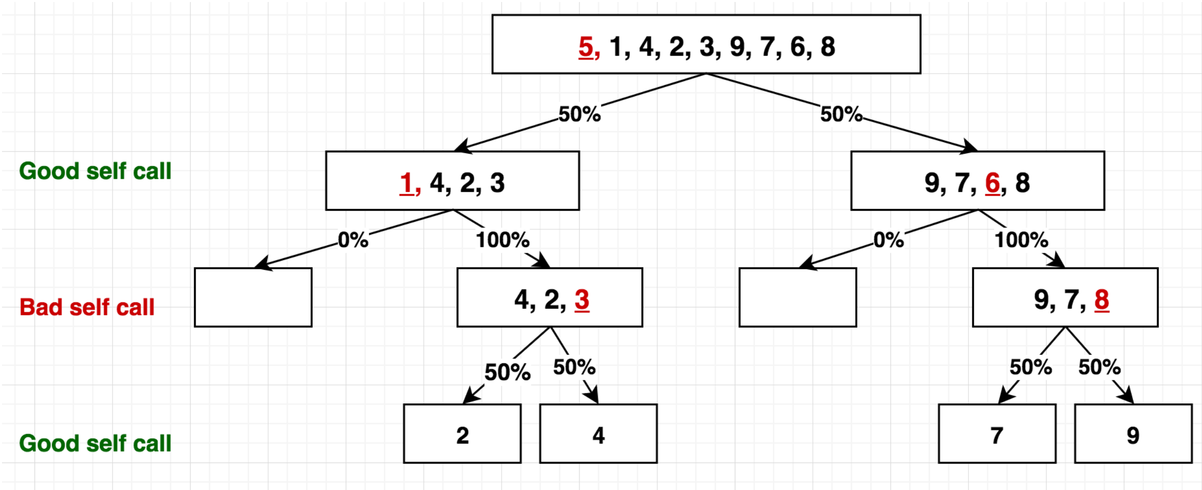

b. {8, 7, 6, 5, 4, 3, 2, 1, 9}




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




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




3. **Question 3** - Illustrate Quick Select

a. {1, 2, 3, 4, 5, 6, 7, 8, 9}, find 5$^{th}$

| | | |
|---|---|---|
| Good self call | 1, 2, 3, 4, 5, 6, 7, 8, 9 => 1, 2, 3,4, 5, 6, 7, 8, 9 | k = 5, k > ILI + IEI, QuickSelect(G, 5 - ILI - IEI) |
| Bad self call | 5, 6, 7, 8, 9 => 5, 6, 7,8, 9 | k = 1, k <= ILI, QuickSelect(L, 1) |
| Good self call | 5, 6, 7 => 5,6, 7 | k = 1, k <= ILI, QuickSelect(L, 1) |
| | 5 | k = 1, ILI < k <= ILI + IEI => 5th = 5 |

b. {8, 7, 6, 5, 4, 3, 2, 1, 9}, find 3$^{th}$

| | | |
|---|---|---|
| Good self call | 8, 7, 6, 5, 4 , 3, 2, 1, 9 => 3, 2, 1, 4 , 8, 7, 6, 9 | k = 3, k <= ILI + IEI, QuickSelect(L, 3) |
| Good self call | 3, 2, 1 => 1, 2, 3 | k = 3, k > ILI + IEI, QuickSelect(G, 3 - ILI - IEI) |
| Good self call | 3 | k = 1, ILI < k <= ILI + IEI => 3th = 3 |

c. {9, 1, 8, 2, 7, 3, 6, 4, 5}, find 8$^{th}$

| | | |
|---|---|---|
| Good self call | 9, 1, 8, 2, 7, 3, 6, 4, 5 => 1, 2, 3, 4, 5, 9, 8, 7, 6 | k = 8, k > ILI + IEI, QuickSelect(G, 8 - ILI - IEI = 8 - 4 -1) |
| Bad self call | 6, 7, 8, 9 => 6, 7, 8, 9 | k = 3, k > ILI + IEI, QuickSelect(G, 3 - ILI - IEI = 3 - 0 - 1) |
| Good self call | 7, 8, 9 => 7, 8, 9 | k = 2, ILI < k <= ILI + IEI => 2th = E{0} = 8 |

d. {5, 1, 4, 2, 3, 9, 7, 6, 8}, find 5$^{th}$

| | | |
|---|---|---|
| Good self call | 5, 1, 4, 2, 3, 9, 7, 6, 8 => 1, 4, 2, 3, 5, 9, 7, 6, 8 | k = 5, ILI < k <= ILI + IEI => 5th = E{0} = 5 |

4. **Question 4** - Redefine "Good Self Call" "Bad Self Call"

If all self-calls are good, height of tree is m = 1 + $\log_{3/2}$ n. So, the expected number of levels to get m self-calls is still *O(logn)*

Therefore, the cost for processing at each level is n, time complexity is *O(n)* and the height is O(logn) => We can conclude that average case running time is *O(nlogn)*. This is no change as compared to ¾ way of division