**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
2. *“Wooden blocks toys”*

Say, Blue block is 1, Red block is 0

of n *0 and 1*

0 at a side, 1 at the other side

This algorithm does not need any additional space, **means in-place algorithm**

Time complexity is *O(n)*

1. *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

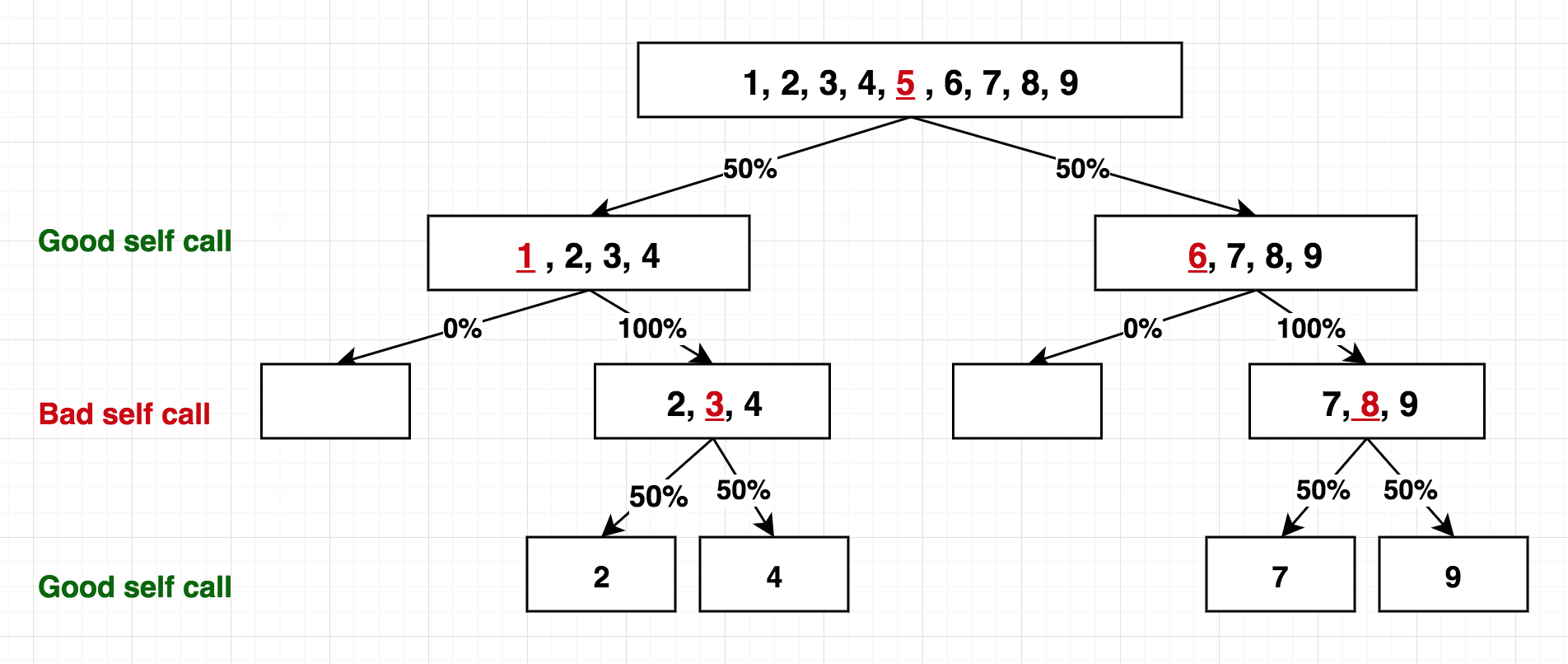
We can simply use Quick Sort algorithm to divide the blocks into 3 parts.

As we all know, Quick Sort algorithm can be done in-place with time complexity is *O(nlogk)(k is number of colors)*

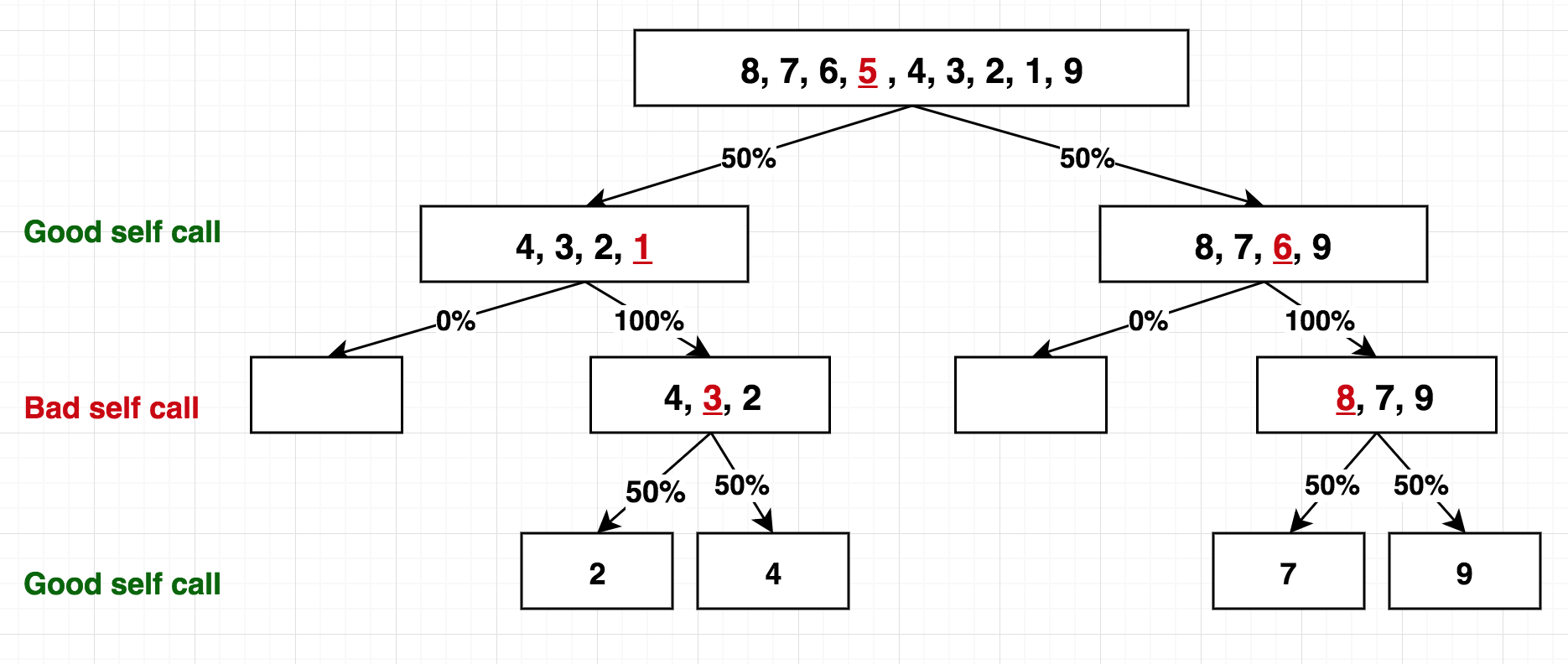
1. *Solve the problem for three different colors*: *Blue, Red, Green* and *Yellow*

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(nlog k)(k is number of colors)*

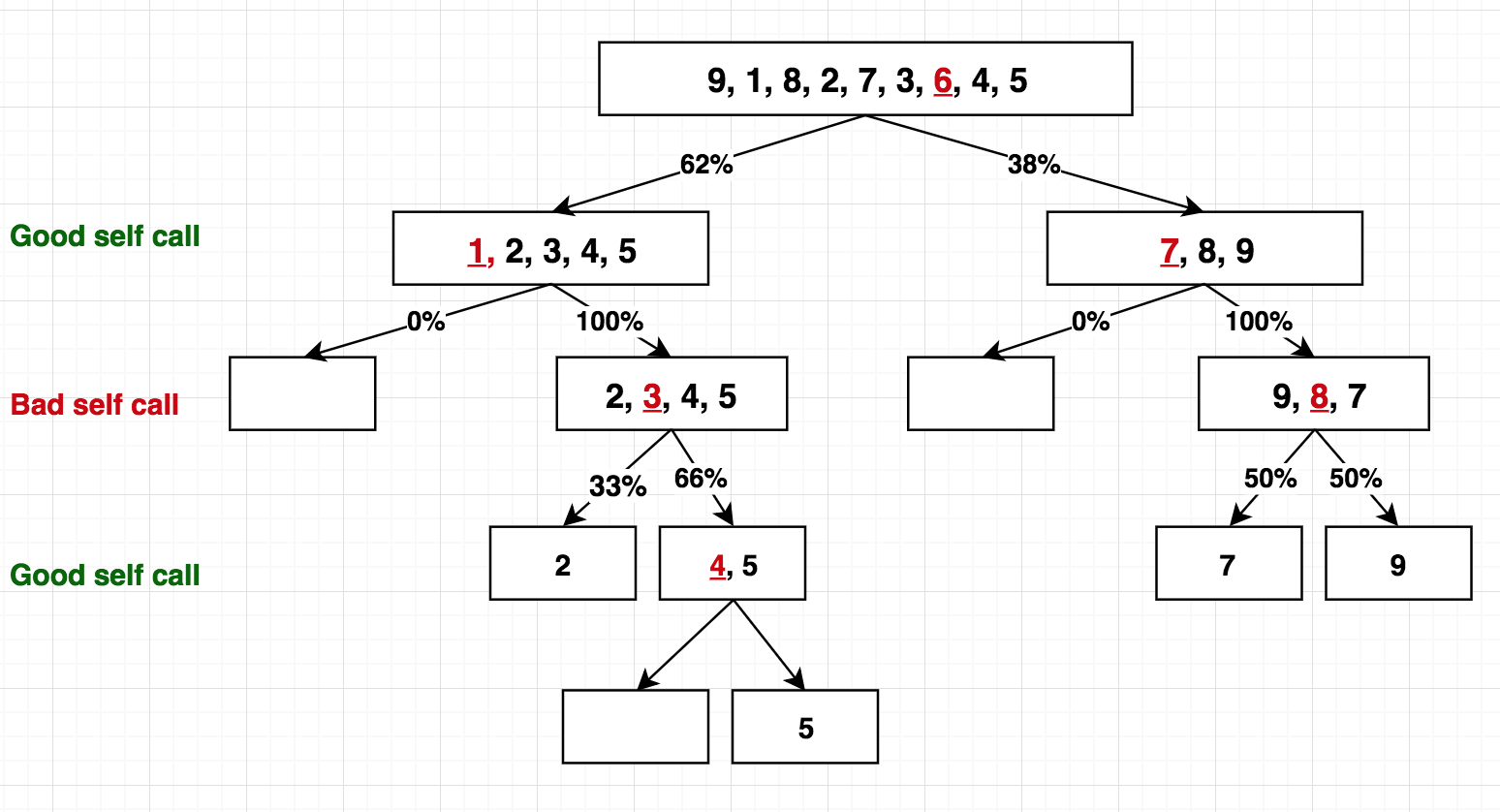
1. **Question 2 -** Illustrate Quick sort, pick a pivot so that they lead to alternating between “Good Self Call” and “Bad Self Call”
2. {1, 2, 3, 4, 5, 6, 7, 8, 9}



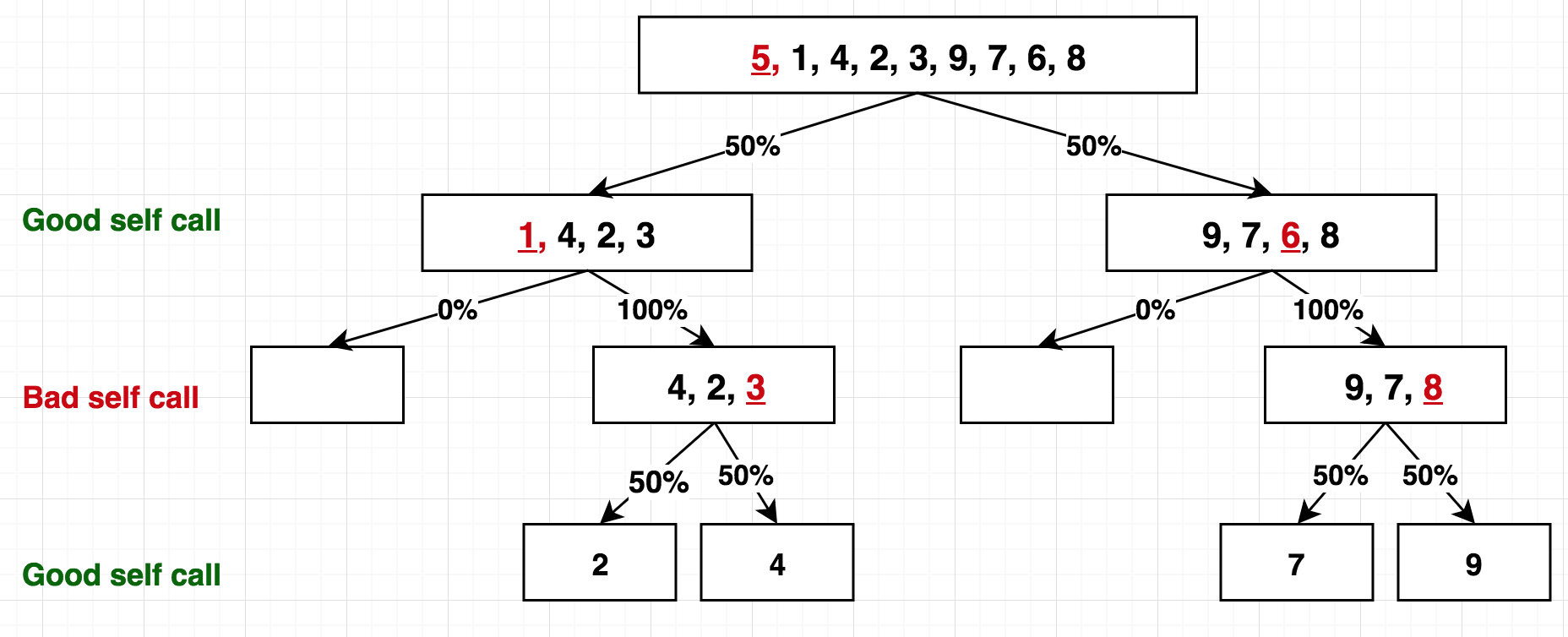
1. {8, 7, 6, 5, 4, 3, 2, 1, 9}



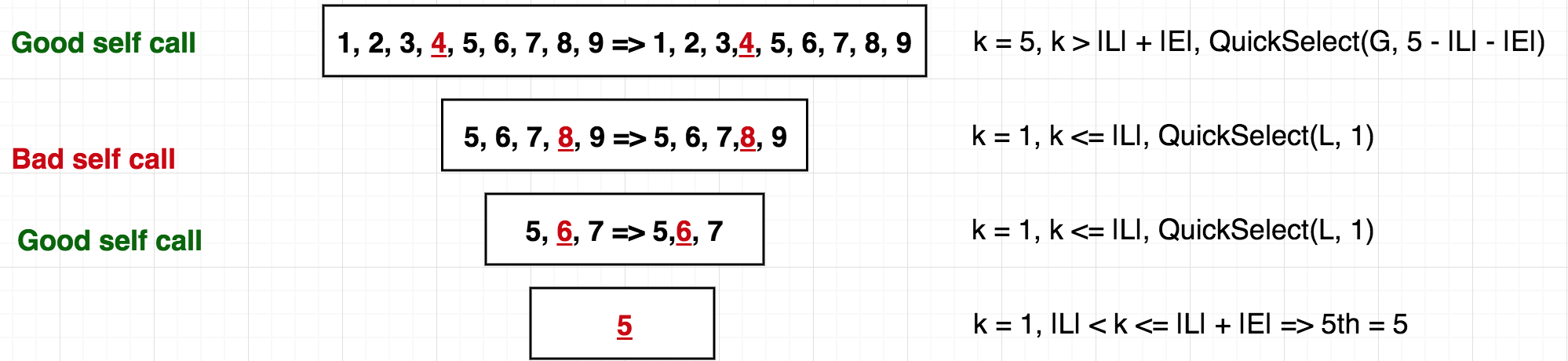
1. {9, 1, 8, 2, 7, 3, 6, 4, 5}



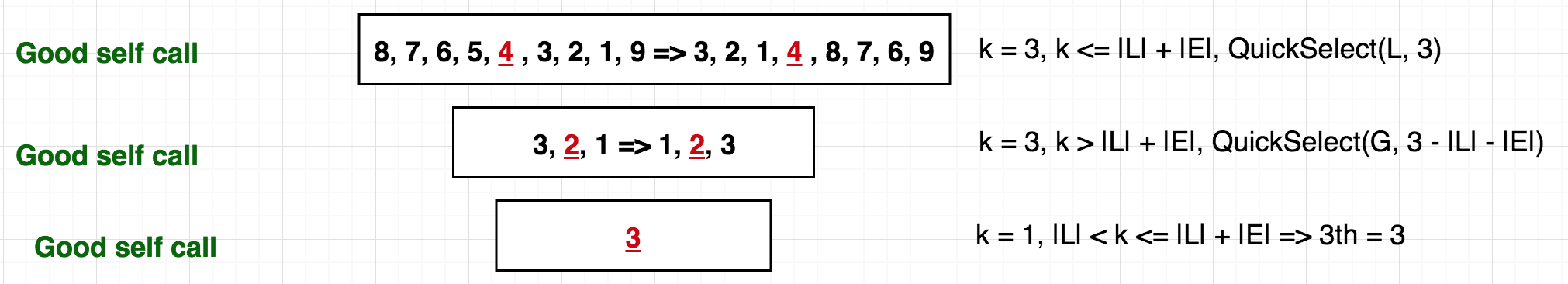
1. {5, 1, 4, 2, 3, 9, 7, 6, 8}



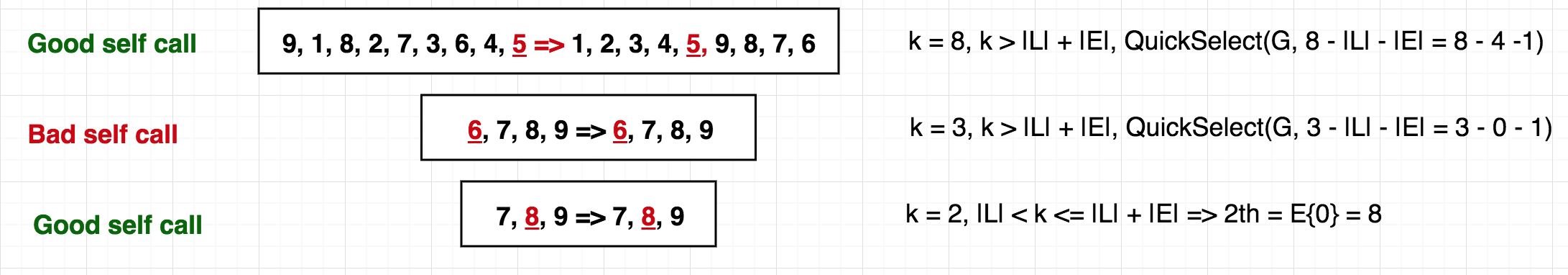
1. **Question 3** - Illustrate Quick Select
2. {1, 2, 3, 4, 5, 6, 7, 8, 9}, find 5th



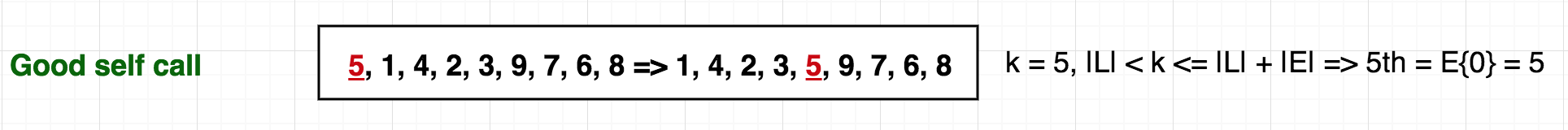
1. {8, 7, 6, 5, 4, 3, 2, 1, 9}, find 3th



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



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



1. **Question 4 -** Redefine “Good Self Call” “Bad Self Call”

If all self-calls are good, height of tree is m = 1 + log3/2 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