Abdelrahman mohamed Rashad

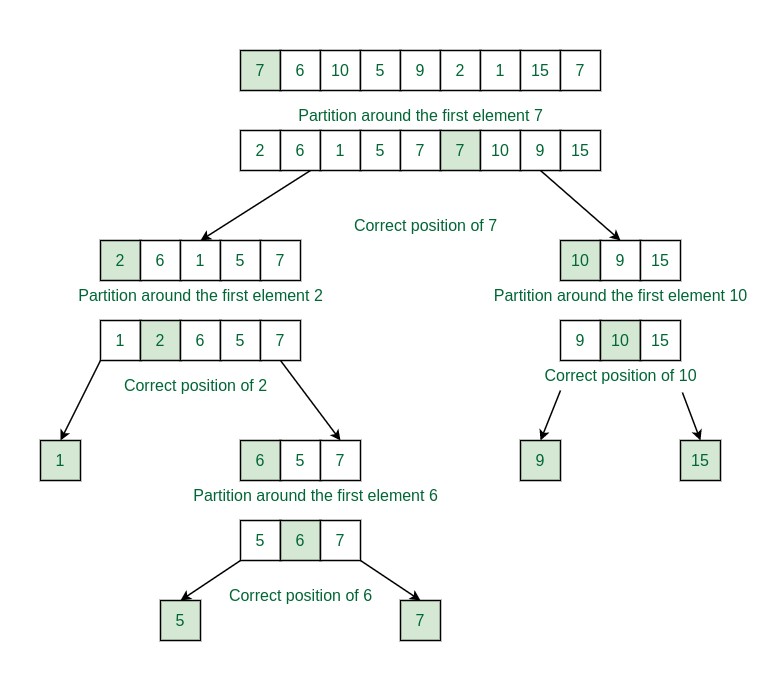
223797

Quick sort

1. Quicksort is divide and conquer algorithem efficient, general-purpose sorting algorithm. Quicksort was developed by British computer scientist Tony Hoare in 1959 and published in 1961. It is still a commonly used algorithm for sorting.

Best complexity: n\*log(n)

Class: Sorting algorithm



1. What are the disadvantages of quicksort?

Quicksort also has some disadvantages, such as being unstable, sensitive to the choice of the pivot, and vulnerable to the worst case. Quicksort can also cause stack overflow if the recursion depth is too high

1. Code :
2. def quicksort(arr):  
    if len(arr) <= 1:  
    return arr  
    else:  
    # Choose pivot using median-of-three  
    pivot = median\_of\_three(arr[0], arr[len(arr)//2], arr[-1])  
     
    # Partition the array  
    less = [x for x in arr if x < pivot]  
    equal = [x for x in arr if x == pivot]  
    greater = [x for x in arr if x > pivot]  
     
    # Recursively sort the partitions  
    return quicksort(less) + equal + quicksort(greater)  
     
   def median\_of\_three(a, b, c):  
    return sorted([a, b, c])[1]  
     
   # Example usage:  
   arr = [3, 1, 4, 52,1, 5, 9, 2, 6, 5,10,111,55, 3, 5]  
   sorted\_arr = quicksort(arr)  
   print(sorted\_arr)

this code is a recursive code using divide and conquer algorithm

1. complexity : *O*(*n*log*n*) on average.

Worst case: *O*(*n*2).

Best case: *O*(*n*log*n*) .

1. space complexity of median of three quick sort is O(log n) which is used for storing call stack formed due to recursion.