

Câu hỏi 4

Chính xác

Chấm điểm của 2,00

Implement static methods **Partition** and **QuickSort** in class **Sorting** to sort an array in ascending order.

```
#ifndef SORTING_H
#define SORTING_H
#include <sstream>
#include <iostream>
#include <type_traits>
using namespace std;
template <class T>
class Sorting {
private:
    static T* Partition(T* start, T* end) ;
public:
    static void QuickSort(T* start, T* end) ;
};
#endif /* SORTING_H */
```

You can read the pseudocode of the algorithm used to in method **Partition** in the below image.

ALGORITHM

HoarePartition(A[l..r])

//Partitions a subarray by Hoare's algorithm, using the first element
// as a pivot
//Input: Subarray of array A[0..n - 1], defined by its left and right
// indices l and r (l < r)
//Output: Partition of A[l..r], with the split position returned as
// this function's value
p ← A[l]
i ← l; j ← r + 1
repeat
repeat i ← i + 1 until A[i] ≥ p
repeat j ← j - 1 until A[j] ≤ p
swap(A[i], A[j])
until i ≥ j
swap(A[i], A[j]) //undo last swap when i ≥ j
swap(A[l], A[j])
return j

For example:

Test	Result
int array[] = { 3, 5, 7, 10 ,12, 14, 15, 13, 1, 2, 9, 6, 4, 8, 11, 16, 17, 18, 20, 19 }; cout << "Index of pivots: "; Sorting<int>::QuickSort(&array[0], &array[20]); cout << "\n"; cout << "Array after sorting: "; for (int i : array) cout << i << " ";	Index of pivots: 2 0 0 6 1 0 2 1 0 0 2 1 0 0 0 0 0 0 1 0 Array after sorting: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Answer: (penalty regime: 0 %)

Reset answer

```
1 static T* Partition(T* start, T* end) {
2     // TODO: return the pointer which points to the pivot afte
3     T pivot = start[0];    // pivot
4     int i = 0;
5     int j = end - start;
6     do {
7         do {
8             i++;
9         }
```

```

10     while (start[i] <= pivot);
11     do {
12         j--;
13     }
14     while (start[j]>pivot);
15
16     swap(start[i], start[j]);
17 }
18 while (i<j);
19
20 swap(start[i], start[j]);
21 swap(start[0], start[j]);
22 return start + j;
23 }
24
25 static void QuickSort(T* start, T* end) {
26     // TODO
27     // In this question, you must print out the index of pivot
28     if (start >= end) return;
29     T* index = Partition(start, end);
30     cout << index - start << " ";
31     QuickSort(start, index);
32     QuickSort(index + 1, end);
33 }

```

Precheck

Kiểm tra

	Test	Expected	Got	
✓	<pre> int array[] = { 3, 5, 7, 10, 12, 14, 15, 13, 1, 2, 9, 6, 4, 8, 11, 16, 17, 18, 20, 19 }; cout << "Index of pivots: "; Sorting<int>::QuickSort(&array[0], &array[20]); cout << "\n"; cout << "Array after sorting: "; for (int i : array) cout << i << " "; </pre>	<pre> Index of pivots: 2 0 0 6 1 0 2 1 0 0 2 1 0 0 0 0 0 0 1 0 Array after sorting: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 </pre>	<pre> Index of pivots: 2 0 0 6 1 0 2 1 0 0 2 1 0 0 0 0 0 0 1 0 Array after sorting: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 </pre>	✓

Passed all tests! ✓

BÁCH KHOA E-LEARNING



WEBSITE

HCMUT

MyBK

BKSI

LIÊN HỆ

📍 268 Lý Thường Kiệt, P.14, Q.10, TP.HCM

☎ (028) 38 651 670 - (028) 38 647 256 (Ext: 5258, 5234)

✉ elarning@hcmut.edu.vn

Copyright 2007-2022 BKEL - Phát triển dựa trên Moodle