Câu hỏi 5

Chính xác

Chấm điểm của 2,00

Implement static methods **Merge** and **MergeSort** in class Sorting to sort an array in ascending order. The Merge method has already been defined a call to method printArray so you do not have to call this method again to print your array.

```
#ifndef SORTING_H
#define SORTING_H
#include <iostream>
using namespace std;
template <class T>
class Sorting {
public:
    /* Function to print an array */
    static void printArray(T *start, T *end)
    {
        long size = end - start + 1;
        for (int i = 0; i < size - 1; i++)
            cout << start[i] << ", ";
        cout << start[size - 1];
        cout << endl;
    }

    static void merge(T* left, T* middle, T* right){
        /*TODO*/
        Sorting::printArray(left, right);
    }
    static void mergeSort(T* start, T* end) {
       /*TODO*/
    }
};
#endif /* SORTING_H */
```

For example:

Test	Result

`int arr[] = {0,2,4,3,1,4};`
`Sorting<int>::mergeSort(&arr[0], &arr[5]);`
	`0, 2`
	`0, 2, 4`
	`1, 3`
	`1, 3, 4`
	`0, 1, 2, 3, 4, 4`

`int arr[] = {1};` `Sorting<int>::mergeSort(&arr[0], &arr[0]);`	

Answer: (penalty regime: 0, 0, 0, 5, 10, 15, ... %)

Reset answer

```
1  static void merge(T* left, T* middle, T* right){
2      /*TODO*/
3      int left_size = middle - left + 1;
4      int right_size = right - middle;
5
6      int leftArr[left_size];
7      int rightArr[right_size];
8
9      for (int i = 0; i < left_size; ++i){
10         leftArr[i] = left[i];
11     }
12
13     for (int j = 0; j < right_size; ++j){
```

```
        rightArr[j] = middle[j + 1];
    }

    int leftIdx = 0;
    int rightIdx = 0;
    int mergedArrIdx = 0;

    while (leftIdx < left_size && rightIdx < right_size) {
        if (leftArr[leftIdx] <= rightArr[rightIdx]) {
            left[mergedArrIdx] = leftArr[leftIdx];
            leftIdx++;
        }
        else {
            left[mergedArrIdx] = rightArr[rightIdx];
            rightIdx++;
        }
        mergedArrIdx++;
    }
    while (leftIdx < left_size) {
        left[mergedArrIdx] = leftArr[leftIdx];
        leftIdx++;
        mergedArrIdx++;
    }
    while (rightIdx < right_size) {
        left[mergedArrIdx] = rightArr[rightIdx];
        rightIdx++;
        mergedArrIdx++;
    }
    Sorting::printArray(left, right);
}
static void mergeSort(T* start, T* end){
    /*TODO*/
    if (start >= end) return;

    T* mid = start + (end - start) / 2;
    mergeSort(start, mid);
    mergeSort(mid + 1, end);
    merge(start, mid, end);
}
```

Precheck | Kiểm tra

	Test	Expected	Got	

✔	`int arr[] = {0,2,4,3,1,4};`
	`Sorting<int>::mergeSort(&arr[0], &arr[5]);`
		`0, 2`
		`0, 2, 4`
		`1, 3`
		`1, 3, 4`
		`0, 1, 2, 3, 4, 4`
			`0, 2`
			`0, 2, 4`
			`1, 3`
			`1, 3, 4`
			`0, 1, 2, 3, 4, 4`
				✔

✔	`int arr[] = {1};` `Sorting<int>::mergeSort(&arr[0], &arr[0]);`			✔

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)

elearning@hcmut.edu.vn