

Câu hỏi 1

Chính xác

Điểm 1,00 của 1,00

🚩 Cờ câu hỏi

Implement method `bubbleSort()` in class `SLinkedList` to sort this list in ascending order. After each bubble, we will print out a list to check (using `printList()`).

```
#include <iostream>
#include <sstream>
using namespace std;

template <class T>
class SLinkedList {
public:
    class Node; // Forward declaration
protected:
    Node* head;
    Node* tail;
    int count;
public:
    SLinkedList()
    {
        this->head = nullptr;
        this->tail = nullptr;
        this->count = 0;
    }
    ~SLinkedList(){};
    void add(T e)
    {
        Node *pNew = new Node(e);

        if (this->count == 0)
        {
            this->head = this->tail = pNew;
        }
```

```
        void add(T e)
        {
            Node *pNew = new Node(e);

            if (this->count == 0)
            {
                this->head = this->tail = pNew;
            }
            else
            {
                this->tail->next = pNew;
                this->tail = pNew;
            }

            this->count++;
        }
        int size()
        {
            return this->count;
        }
        void printList()
        {
            stringstream ss;
            ss << "[";
            Node *ptr = head;
            while (ptr != tail)
            {
                ss << ptr->data << ", ";
                ptr = ptr->next;
            }

            if (count > 0)
                ss << ptr->data << "]";
```

Đi

1

Hil

Hc

Điều

1

Hiện th

Hoàn tl

```

        while (ptr != tail)
        {
            ss << ptr->data << ",";
            ptr = ptr->next;
        }

        if (count > 0)
            ss << ptr->data << "]";
        else
            ss << "]";
        cout << ss.str() << endl;
    }
public:
    class Node {
    private:
        T data;
        Node* next;
        friend class SLinkedList<T>;
    public:
        Node() {
            next = 0;
        }
        Node(T data) {
            this->data = data;
            this->next = nullptr;
        }
    };

    void bubbleSort();
};

```

```
};
```

For example:

Test	Result
int arr[] = {9, 2, 8, 4, 1};	[2,8,4,1,9]
SLinkedList<int> list;	[2,4,1,8,9]
for(int i = 0; i <int(sizeof(arr))/4;i++)	[2,1,4,8,9]
list.add(arr[i]);	[1,2,4,8,9]
list.bubbleSort();	

Answer: (penalty regime: 0 %)

Reset answer

Câu hỏi 2

Chính xác

Điểm 1,00 của 1,00

🚩 Cờ câu hỏi

Implement static method **selectionSort** in class **Sorting** to sort an array in ascending order. After each selection, we will print out a list to check (using **printArray**).

```
#include <iostream>
using namespace std;

template <class T>
class Sorting
{
public:
    /* Function to print an array */
    static void printArray(T *start, T *end)
    {
        int size = end - start;
        for (int i = 0; i < size - 1; i++)
            cout << start[i] << " ";
        cout << start[size - 1];
        cout << endl;
    }

    static void selectionSort(T *start, T *end);
};
```

For example:

Test	Result
int arr[] = {9, 2, 8, 1, 0, -2};	-2, 2, 8, 1, 0, 9
Sorting<int>::selectionSort(&arr[0], &arr[6]);	-2, 0, 8, 1, 2, 9
	-2, 0, 1, 8, 2, 9
	-2, 0, 1, 2, 8, 9
	-2, 0, 1, 2, 8, 9

Câu hỏi 3

Chính xác

Điểm 1,00 của 1,00

🚩 Cờ câu hỏi

Implement static methods **sortSegment** and **ShellSort** 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 void printArray(T* start, T* end)
    {
        int size = end - start;
        for (int i = 0; i < size; i++)
            cout << start[i] << " ";
        cout << endl;
    }

public:
    // TODO: Write your code here
    static void sortSegment(T* start, T* end, int segment_idx, int cur_segment_total);
    static void ShellSort(T* start, T* end, int* num_segment_list, int num_phases);
};

#endif /* SORTING_H */
```

For example:

Test	Result
int num_segment_list[] = {1, 3, 5};	5 segments: 5 4 3 2 1 10 9 8 7 6
int num_phases = 3;	3 segments: 2 1 3 5 4 7 6 8 10 9
int array[] = { 10, 9, 8 , 7 , 6, 5, 4, 3, 2, 1 };	1 segments: 1 2 3 4 5 6 7 8 9 10
Sorting<int>::ShellSort(&array[0], &array[10], &num_segment_list[0], num_phases);	

Answer: (nanatu ranima: 0 %)

Câu hỏi 4

Chính xác
Điểm 1.00 của 1.00
Cờ câu hỏi

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[i], 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

For example:

	Result
12, 14, 15, 13, 1, 2, 9, 6, 4, 8, 11, 16, 17, 18, 20, 19 } ; y[0], &array[20]); "; << " ";	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



