Thời gian còn lại 0:19:30

Câu hỏi 13

Chính xác

Điểm 1,00 của 1,00

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

Answer: (penalty regime: 0 %)

Reset answer

```
template <class T>
void SLinkedList<T>::bubbleSort() {
    if (count <=1)  {return;}
    bool cmp = true;
    for (int i=0;i<count;i++)
    {
        cmp = false;
        Node* curr = head->next;
        Node* prev = head;
        for (; curr != NULL; prev = curr, curr= curr->next)
        {
            if (prev->data > curr->data)
            {
                T temp = prev->data;
                prev->data = curr->data;
                curr->data = temp;
                cmp = true;
            }
        }
        if (cmp==true)
            printList();
    }
}
```

Precheck Kiểm tra

	Test	Expected	Got	

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

Passed all tests! ✓

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 14

Chính xác

Điểm 1,00 của 1,00

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

Answer: (penalty regime: 0 %)

Reset answer

```
template <class T>
void Sorting<T>::selectionSort(T *start, T *end)
{
    int size = end - start;
    for (int i=0;i<size-1;i++)
    {
        int minIdx = i;
        for (int j=i+1;j<size;j++)
        {
            if (start[j]< start[minIdx])
            {
                minIdx=j;
            }
        }
        if  (minIdx!=i)
        {
            T temp = start[i];
            start[i] = start[minIdx];
            start[minIdx] = temp;
        }
        printArray(start,end);
    }
}
```

Precheck Kiểm tra

	Test	Expected	Got	

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

Passed all tests! ✓

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 15

Precheck results

Điểm 0,00 của 1,00

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

Answer: (penalty regime: 0 %)

Reset answer

```
// TODO: Write your code here

static void sortSegment(T* start, T* end, int segment_idx, int cur_segment_total) { //insertion sort
    // TODO
    int segment_size = (end-start) / cur_segment_total;
    T* segment_start = start + segment_idx*segment_size;
    T* segment_end = segment_start + segment_size;


    return ;
}

static void ShellSort(T* start, T* end, int* num_segment_list, int num_phases) {
    // TODO
    // Note: You must print out the array after sorting segments to check whether your algorithm is true.
    for (int i=num_phases-1;i>=0;i--)
    {
        int k=num_segment_list[i];
        for (int segment_idx=0; segment_idx < k; segment_idx++)
```

```
            sortSegment(start, end, segment_idx, k);
        cout << k <<" segments: ";
        printArray(start,end);
    }
}
```

Precheck   Kiểm tra

Precheck only

	Test	Expected	Got	

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

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