Cấu trúc dữ liệu và giải thuật

Cuối kì
Harmony BTL1 + BTL2

nhóm thảo luận CSE
https://www.facebook.com/groups/211867931379013

Tp. Hồ Chí Minh, Tháng 10/2023

Mục lục

1 BTL 1

1. [harmony BTL1] Điền code vào ô còn trống Đang hiện thực danh sách liên kết vòng đôi với nút đang quản lí danh sách liên kết vòng đôi là *tail*, điền vào chỗ (1), (2), ...

```
template <typename E> class Node
{
public:
    E element;
    Node *next, *prev;
public:
    Node(): next(nullptr), prev(nullptr) {}
};
template <typename E> void insert (Node<E>*& tail, Node<E> *newNode, int
↪   index)
{
    int size = len(tail); //! kích thước của DSLK
    if(size == 0){
        /*(1)*/
        tail = newNode;
    }
    // insert tail
    else if(index >= size)
    {

        /*(2)*/
        /*(3)*/
        newNode = tail;
    }
    // insert mid
    else
    {
        Node<E> temp = tail;
        while(index > 0){
            temp = temp->next;
            index --;
        }
        /*(4)*/
        /*(5)*/
        /*(6)*/
        /*(7)*/
    }
}
```

Solution:
(1) : newNode->next = newNode->prev = newNode;

(2) : tail->next = newNode;
(3) : newNode->prev = tail;
(4) : temp->next->prev = newNode;
(5) : newNode->next = temp->next;
(6) : newNode->prev = temp;
(7) : temp->next = newNode;

2. [harmony BTL1] giống câu 1 nhưng chỉnh $index > 0$ thành $index > -1$

Solution:
(1) : newNode->next = newNode->prev = newNode;
(2) : tail->next = newNode;
(3) : newNode->prev = tail;
(4) : temp->prev->next = newNode;
(5) : newNode->next = temp;
(6) : newNode->prev = temp->prev;
(7) : temp->prev = newNode;

3. [harmony BTL1] Điền code vào ô còn trống Đang hiện thực danh sách liên kết vòng đôi với nút đang quản lí danh sách liên kết vòng đôi là *tail*, điền vào chỗ (1), (2), ...

```
template <typename E> class Node
{
private:
    E element;
    Node *next, *prev;
public:
    Node(): next(nullptr), prev(nullptr) {}
};

template <typename E> void remove(Node<E>*& tail, E element)
{
    int size = len(tail); //! kích thước của DSLK
    if(size == 1) delete tail;
    // delte tail
    else if(tail->element == element)
    {
        Node<E>* delN = tail;
        /*(1)*/
        /*(2)*/
        /*(3)*/
        delete delN;
    }
    else
    {
        Node<E> temp = tail->next;
        while(temp != tail)
        {
            if(temp->element == element)
            {
                Node<E>* delN = temp;
                /*(4)*/
                /*(5)*/
                /*(6)*/
                break;
            }
            /*(7)*/
        }
    }
}
```

Solution:
(1) : tail->next->prev = tail->prev;
(2) : tail->prev->next = tail->next;

(3) : tail = tail->prev;
(4) : temp->next->prev = temp->prev;
(5) : temp->prev->next = temp->next;
(6) : delete delN;
(7) : temp = temp->next;

4. [harmony BTL1] Điền code vào ô còn trống Đang hiện thực danh sách liên kết vòng đôi với nút đang quản lí danh sách liên kết vòng đôi là *tail*, điền vào chỗ (1), (2), ..., với hàm REVERSAL có chức năng revese các khách hàng oán linh với nhau không bao gồm khách hàng thuật sư.

```
template <typename E> class Node
{
private:
    E element;
    Node *next, *prev;
public:
    Node(): next(nullptr), prev(nullptr) {}
};
//! trả về oán linh tiếp theo chiều thuận đồng hồ, nếu không tìm thấy trả
↪  về null, có thể là chính nó
template <typename E> Node<E> getNextSpirit(Node<E>* node);
//! trả về oán linh trước đó theo chiều nghịch kiem đồng hồ, nếu không
↪  tìm thấy trả về null, có thể là chính nó
template <typename E> Node<E> getNPrevSpirit(Node<E>* node);

template <typename E> void REVERSAL(Node<E>*& tail)
{
    if(tail == nullptr) return;
    Node<E>* headSpirit = getNextSpirit(tail->next);
    Node<E>* tailSpirit = /*(1)*/;

    // code đã kiểm tra nếu số khách oán linh <= 1 thì return tail
    if(/*(0)*/) return;

    while(headSpirit != tailSpirit)
    {
        swap(/*(2)*/);
        headSpirit = /*(3)*/;

        if(headSpirit == tailSpirit) return;
        tailSpirit = /*(4)*/;
    }
}

```

Solution:
(0) : headSpirit == tailSpirit
(1) : getNPrevSpirit(tail)
(2) : headSpirit->element, tailSpirit->element
(3) : getNextSpirit(headSpirit->next)
(4) : getNPrevSpirit(tailSpirit->prev)

5. [harmony BTL1] Cho danh sách hàng chờ của nhà hàng theo thứ tự của khách là {name="A", energy = 4}, {name="C", energy = -1}, {name="B", energy = 1}, {name="D", energy = 4}, {name="E", energy = 2} , {name="K", energy = 3}, {name="H", energy = 3}, {name="X", energy = 2}, {name="Y", energy = 2}, với danh sách thời gian giống danh sách hàng chờ với khách hàng vô đầu tiên là A và khách hàng mới vô là Y, hãy dùng giải thuật *sellsort* với bước nhảy là 1, 2, 3, .. $n/2$ xếp khách hàng theo $abs(ereny)$ giảm dần với khách hàng mới vô sẽ xếp trước khách hàng vô sau đó nếu cùng $abs(ereny)$ (stable của sell sort), hãy trình bày ra từng bước

Solution:
bước 1, bước nhảy = 4 :
bước 2, bước nhảy = 2 :
bước 3, bước nhảy = 1 :

nhóm thảo luận CSE

https://www.facebook.com/groups/211867931379013

CHÚC CÁC EM THI TỐT