

Môn thi: Cấu trúc dữ liệu và giải thuật
Mã lớp: IT003 – Hệ đại trà
Thời gian làm bài: 90 phút
(Sinh viên không được sử dụng tài liệu)

Câu 1: (2 điểm)

Cho cấu trúc dữ liệu danh sách liên kết đơn và hàm Rprint như dưới đây:

<pre>struct Node { int info; Node *Next; };</pre>	<pre>struct List { Node *Head; Node *Tail; };</pre>
---	---

Hàm void Rprint(List ls)

Anh / chị hãy viết hàm Rprint để in lên màn hình các phần tử trong danh sách ls theo thứ tự từ phần tử cuối đến phần tử đầu (**không dùng mảng phụ hoặc danh sách phụ**).

 **Đáp án tham khảo:**

```
void Rprint(List ls) {
    if (ls == NULL)
        cout << "Danh sach rong." ;
    else {
        node *p = ls.Head;
        if (ls.Head != ls.Tail) {
            ls.Head = p->Next;
            Rprint(ls);
        }
        cout << p->info << " ";
    }
}
```

Câu 2: (2 điểm)

Cho chuỗi các thao tác như sau:

RTU*IMUN***EB***1*ON**

Biết rằng: với mỗi chữ cái tượng trưng cho thao tác thêm chữ cái tương ứng vào **Stack (ngăn xếp)**; với mỗi dấu * tượng trưng cho thao tác lấy nội dung một phần tử trong stack và in ra màn hình. Anh / chị hãy cho biết sau khi hoàn tất chuỗi thao tác trên, những chữ cái nào còn trong stack và những chữ cái nào được in ra màn hình.

 **Đáp án tham khảo:**

- Chữ cái in ra màn hình: U I T N U M B E R 1
- Chữ cái còn lại trong Stack:

N
O

Câu 3: (4 điểm)

Cho dãy số như sau: {93, 11, 97, 65, 27, 70, 13, 83, 54, 101}

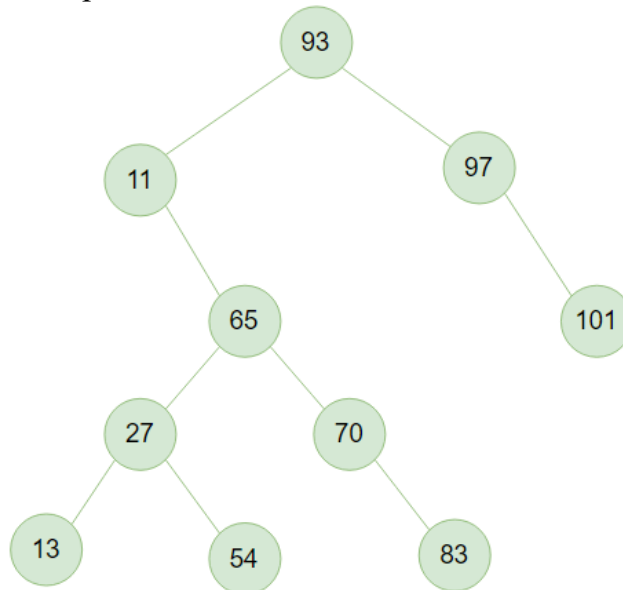
Anh / chị hãy thực hiện:

- Xây dựng **cây nhị phân tìm kiếm** từ dãy số trên **lần lượt từ trái sang phải** (1.5 điểm)
- Xóa lần lượt theo thứ tự các nút {54, 11, 93}. Khi xóa 1 nút, cân bằng cây khi xảy ra mất cân bằng, cho biết nút bị mất cân bằng, loại mất cân bằng. (1.5 điểm)
- Viết hàm đếm số nút trên cây chỉ có duy nhất 1 nút con phải (1 điểm)



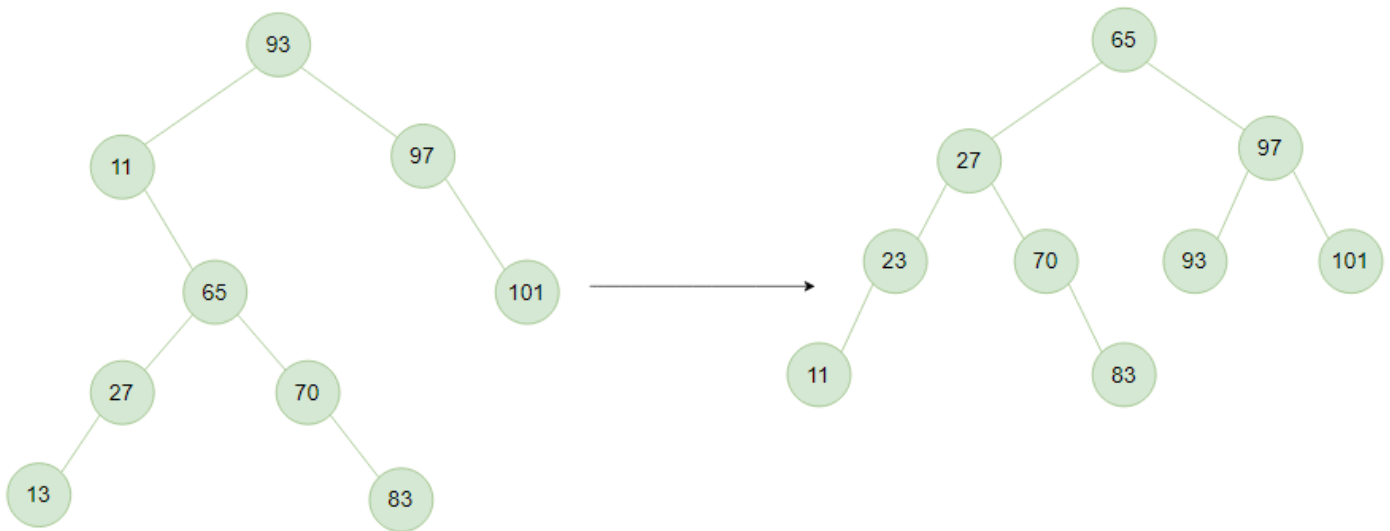
Đáp án tham khảo:

- Cây nhị phân tìm kiếm là cây nhị phân thỏa mãn các nguyên tắc bố trí mỗi khóa, mỗi nút sao cho:
 - Các nút của cây con bên trái nhỏ hơn nút hiện hành.
 - Các nút của con bên phải lớn hơn nút hiện hành.

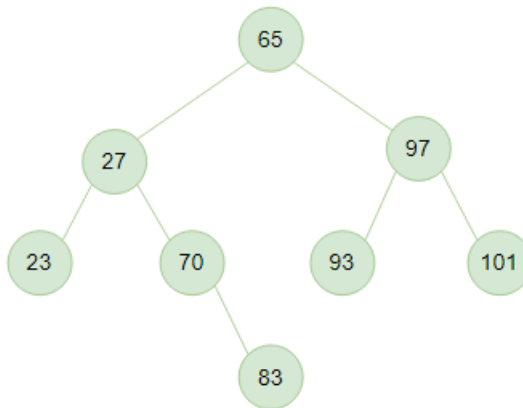


- Cách trình bày lời giải đáp được chấp nhận: (chọn 1 trong 3)
 - Cân bằng cây ở câu 3a sau đó xóa 3 số/node
 - Áp dụng qui tắc cân bằng cây nhị phân tìm kiếm cân bằng khi xóa 3 số/node***
 - Duyệt cây lấy ra các số/node sau đó xây cây nhị phân tìm kiếm cân bằngCuối cùng, xây dựng được cây cân bằng AVL với chuỗi số đã cho (đã loại bỏ 3 số/node)
Lưu ý:
 - Vẽ được cây cân bằng cuối cùng và có lập luận hợp lý (1.5 điểm)
 - Vẽ được cây cân bằng cuối cùng nhưng không trình bày lí do vì sao có cây kết quả (-0.75 điểm)**Xóa 54:** 54 là nút lá → xóa 54 mà không ảnh hưởng đến quy tắc cây nhị phân tìm kiếm; cây AVL mất cân bằng L – R tại nút 93, thực hiện phép quay trái → 11 trở thành cây con trái của 65; cây vẫn còn mất cân bằng L – L, thực hiện phép

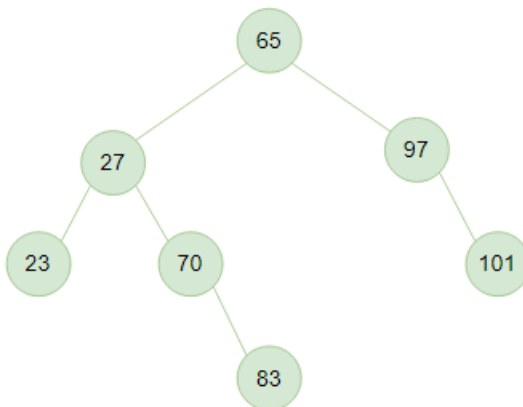
quay phải → 93 trở thành cây con bên phải của 65. Cây tiếp tục mất cân bằng R – L tại nút 65, thực hiện phép quay trái → 97 là gốc mới của 93 và 101. Cây đã cân bằng.



Xóa 11: 11 là nút lá → thực hiện xóa 11, cây vẫn chưa mất cân bằng



Xóa 93: 93 là nút lá → thực hiện xóa, ta được cây AVL kết quả.



```
c. void DemNut(TREE* t) {
    if ( t == NULL)
        return 0;
```

```

else if (t->left == NULL && t->right != NULL)
    return 1 + count(t->left) + count(t->right);
else return count(p->left) + count(p->right);
}

```

Câu 4: (2 điểm)

Cho tập khóa $K = \{12, 44, 13, 88, 23, 94, 11, 39, 20, 16, 5\}$ và hàm băm:

$$H(\text{key}) = (2 * \text{key} + 5) \% 11 \quad (\text{key: khóa cần băm})$$

Anh / chị hãy thực hiện:

- Vẽ hình từng bước việc lưu trữ từng khóa trong K vào bảng băm có kích thước $M = 11$, dùng hàm băm H và phương pháp nối kết trực tiếp để xử lý xung đột (1 điểm)
- Định nghĩa cấu trúc dữ liệu cho bảng băm ở câu a. (1 điểm)

Đáp án tham khảo:

- Hướng giải quyết khi đụng độ: các khóa bị băm có cùng địa chỉ (đụng độ/xung đột) sẽ được gom thành một danh sách liên kết.

Thêm 12: $H(12) = (2.12 + 5) \% 11 = 9$.

Thêm 44: $H(44) = (2.44 + 5) \% 11 = 5$

Thêm 13: $H(13) = (2.13 + 5) \% 11 = 9$

Thêm 88: $H(88) = (2.44 + 5) \% 11 = 5 \rightarrow$ đụng độ

Thêm 23: $H(23) = (2.23 + 5) \% 11 = 7 \rightarrow$ đụng độ

Thêm 94: $H(94) = (2.94 + 5) \% 11 = 6$

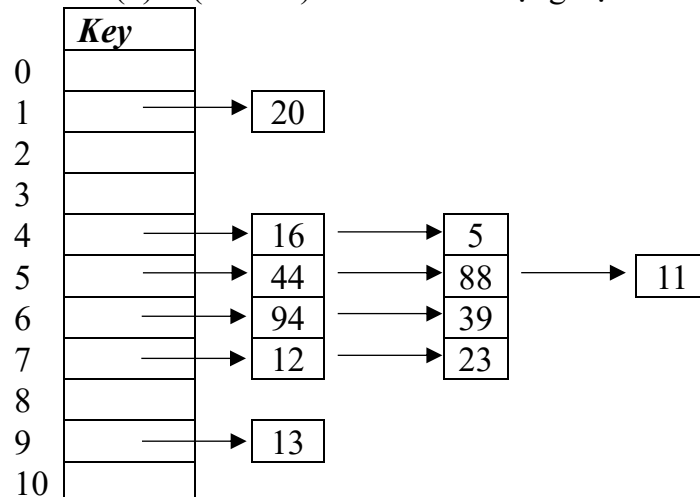
Thêm 11: $H(11) = (2.11 + 5) \% 11 = 5 \rightarrow$ đụng độ

Thêm 39: $H(39) = (2.39 + 5) \% 11 = 6 \rightarrow$ đụng độ

Thêm 20: $H(20) = (2.20 + 5) \% 11 = 1$

Thêm 16: $H(16) = (2.16 + 5) \% 11 = 4$

Thêm 5: $H(5) = (2.5 + 5) \% 11 = 4 \rightarrow$ đụng độ



- ```

#define M 11
typedef struct tagNode {
 int key;
 tagNode *next;
} node, *nodeptr;

```

```
nodeptr h_table[M];
```

**HẾT**