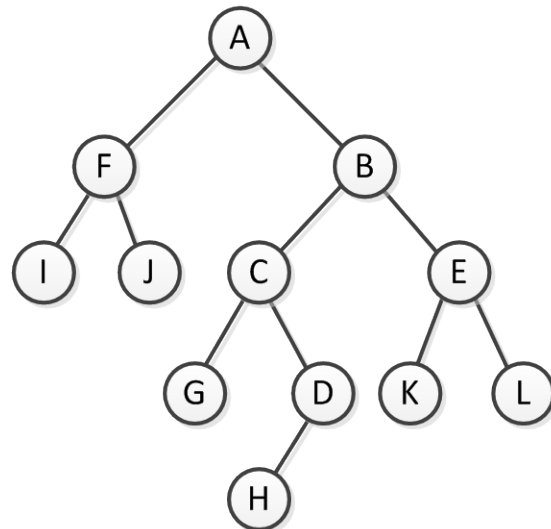
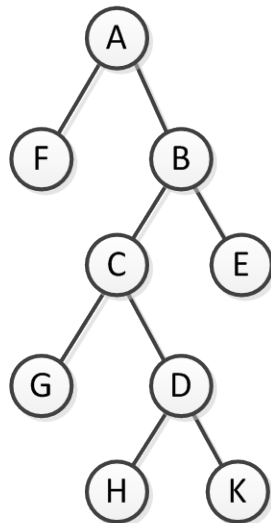
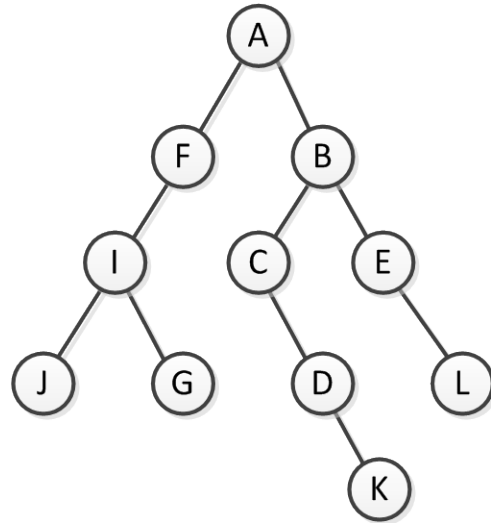
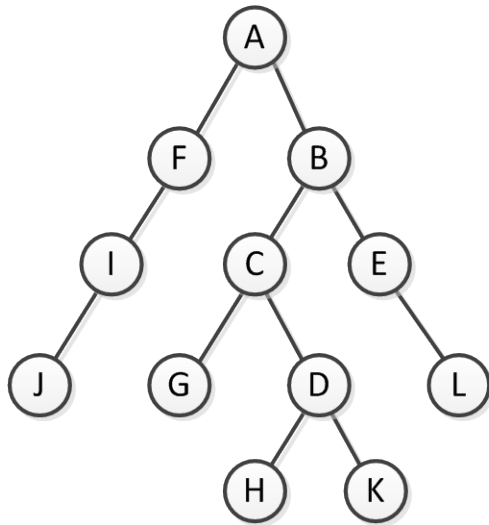


Bài tập chương 4

Phần AVL, Splay, 2-3, Red-Black Tree

Bài 1. Trong các cây nhị phân sau, cây nào là cây AVL, cây nào không phải. Với các cây không phải là AVL, hãy chỉ ra các đỉnh vi phạm.



Bài 2. Số nút tối thiểu trên cây nhị phân cân bằng AVL có chiều cao h là bao nhiêu?

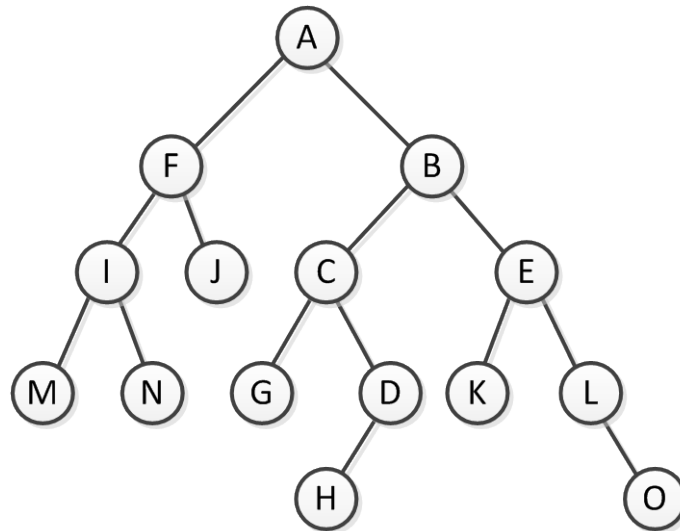
Bài 3. Vẽ cây AVL tạo thành bằng cách thêm lần lượt các khóa sau (vẽ cây sau mỗi lần 1 khóa được thêm vào)

- 3, 5, 7, 23, 12, 76, 87, 54, 19, 4
- D, H, A, G, T, C, D, I, Q, V, B, E
- 12, 34, 53, 76, 15, 21, 18, 45, 16, 55, 11

Bài 4. Với các cây AVL kết quả trong bài 3, hãy vẽ cây thu được sau khi ta xóa lần lượt các nút đã thêm vào theo

- Thứ tự các nút được thêm vào trước sẽ bị xóa trước (FIFO)
- Thứ tự các nút được thêm vào sau sẽ bị xóa trước (LIFO)

Bài 5. Vẽ cây thu được sau khi xóa các nút sau từ cây AVL ở dưới



- | | |
|------|------|
| a. A | d. E |
| b. F | e. C |
| c. B | f. D |

Bài 6. Xây dựng hàm để tìm và trả về chiều cao trên cây AVL, bằng cách đi từ gốc đến 1 nút lá thay vì đến tất cả nút lá

Gợi ý: dựa vào thông tin về trạng thái cân bằng của nút

Bài 7. Xây dựng hàm để tìm và trả về con trỏ tới nút bên trái nhất của cây con phải của một cây AVL khác rỗng

Bài 8. Chứng minh rằng số lượng các phép xoay khi thực hiện xóa một nút trên cây không vượt quá $\frac{1}{2}$ chiều cao của cây

Bài 9. Viết hàm để xóa một nút trên cây AVL

Bài 10. Hoàn thiện chương trình mô phỏng cây tìm kiếm nhị phân cân bằng AVL

Bài 11. Giả sử chúng ta có một dãy số liên tiếp a_1, \dots, a_n và chúng ta cần trả lời nhanh cho câu hỏi dạng: cho i, j hãy tìm và trả về giá trị lớn nhất trong đoạn a_i, \dots, a_j

- Hãy xây dựng cấu trúc dữ liệu để lưu trữ dãy với chi phí $O(n^2)$ về bộ nhớ nhưng thời gian tìm và trả lời câu hỏi chỉ là $O(1)$
- Xây dựng cấu trúc mà chỉ cần chi phí bộ nhớ cỡ $O(n)$, và thời gian tìm và trả lời cho câu hỏi là $O(\log n)$

Bài 12. Cho mảng các số thực gồm n phần tử $A[1, \dots, n]$. Hãy thiết kế thuật toán để thực hiện các công việc sau:

- $Add(i, y)$ cộng giá trị y vào phần tử thứ i trong dãy
- $Partial_sum(i)$ trả về tổng của i phần tử đầu tiên trong dãy $\sum_{j=1}^i a_j$

Các thao tác trên cần có thời gian thực hiện cỡ $O(\log n)$. Ở đây ta không phải thêm hay xóa các phần tử trong dãy, chỉ thực hiện cộng giá trị các phần tử. Có thể sử dụng thêm bộ nhớ phụ nếu cần.

Bài 13. So sánh ưu nhược điểm của các cấu trúc cây AVL, Splay và cây 2-3.

Bài 14. Vẽ cây Splay thu được khi ta thực hiện thêm lần lượt các khóa sau vào cây ban đầu rỗng

- 12, 45, 23, 76, 3, 47, 12, 15, 32
- A, B, D, C, G, H, F, E
- A, B, C, D, E, F, G, H, I, J
- 15, 13, 12, 11, 9, 8, 7, 5, 1

Bài 15. Vẽ cây 2-3 thu được khi ta thực hiện thêm lần lượt các khóa sau vào cây ban đầu rỗng

- 12, 45, 23, 76, 3, 47, 12, 15, 32
- A, B, D, C, G, H, F, E
- A, B, C, D, E, F, G, H, I, J
- 15, 13, 12, 11, 9, 8, 7, 5, 1