


Môn thi: Cấu trúc dữ liệu và Giải thuật  
Mã lớp: IT003 – Chương trình tài năng  
Thời gian làm bài: 90 phút  
(Sinh viên không được sử dụng tài liệu)

**Câu 1 (3 điểm):**

Cho mảng số nguyên **A**: {16, 21, 45, 8, 11, 53, 3, 26, 49, 31, 12}

- Trình bày các bước chính (về từng bước) áp dụng Quicksort để sắp xếp **A** tăng dần. Trong đó, phần tử pivot (còn gọi là nút trục hay phần tử chốt) **x** là phần tử đầu tiên của mảng/mảng con đang được sắp xếp. (2 điểm)
- Theo anh/chị, việc chọn phần tử pivot theo các cách khác nhau như: chọn phần tử đầu tiên, phần tử cuối cùng, phần tử trung vị có ảnh hưởng đến hiệu quả của Quicksort không? Nếu có, cho ví dụ minh họa. (0,5 điểm)
- Cho mảng số nguyên **B**: {2, 3, 4, 5, 6, 7, 8, 9, 1}, anh/chị hãy đề xuất một giải thuật sắp xếp hiệu quả nhất để sắp xếp **B** tăng dần. Giải thích lý do đề xuất. (0,5 điểm)

 **Đáp án tham khảo:**

- Mảng **A** gồm  $n = 11$  phần tử

16	21	45	8	11	53	3	26	49	31	12
----	----	----	---	----	----	---	----	----	----	----

Bước 1: Chọn phần tử 16 làm pivot; phân hoạch mảng **A** thành 2 phần, phần bên trái gồm các phần tử nhỏ hơn 16, phần bên phải gồm các phần tử lớn hơn 16

8	11	3	12	16	21	45	53	26	49	31
---	----	---	----	----	----	----	----	----	----	----

Bước 2: Tiếp tục áp dụng áp dụng Quick Sort cho hai phần của mảng. Chọn phần tử 8 làm pivot cho phần nhỏ hơn {16}

3	8	11	12	16	21	45	53	26	49	31
---	---	----	----	----	----	----	----	----	----	----

Bước 3: Chọn 11 làm pivot cho phần bên phải của {8}, không cần sắp xếp phần bên phải {8} vì chỉ còn lại một phần tử.

3	8	11	12	16	21	45	53	26	49	31
---	---	----	----	----	----	----	----	----	----	----

Bước 4: Tiếp tục áp dụng Quick Sort cho phần lớn hơn {16}, chọn phần tử 21 làm pivot.

3	8	11	12	16	21	45	53	26	49	31
---	---	----	----	----	----	----	----	----	----	----

Bước 5: Bên trái {21} không còn phần tử để sắp xếp. Chọn phần tử 45 làm pivot

3	8	11	12	16	21	26	31	45	49	53
---	---	----	----	----	----	----	----	----	----	----

Bước 6: Chọn phần tử 26 làm pivot cho phân hoạch bên trái {45}

3	8	11	12	16	21	26	31	45	49	53
---	---	----	----	----	----	----	----	----	----	----

Bước 7: Chọn phần tử 49 làm pivot cho phân hoạch bên phải {45}.

3	8	11	12	16	21	26	31	45	49	53
---	---	----	----	----	----	----	----	----	----	----

Bước 8: Kết thúc thuật toán do các phần chỉ còn 1 phần tử không thể phân hoạch được nữa; ta được dãy kết quả đã được sắp xếp tăng dần.

3	8	11	12	16	21	26	31	45	49	53
---	---	----	----	----	----	----	----	----	----	----

- b) Xét mảng A đã được sắp xếp theo thứ tự tăng dần: {1,2,3,4,5,6,7,8,9,10}
- **Trường hợp 1:** Chọn phần tử đầu tiên làm pivot, tức là '1' và áp dụng Quick Sort, ta thấy phần bên trái (các phần tử nhỏ hơn 1) là mảng rỗng. Quá trình phân hoạch này không làm giảm đáng kể kích thước của mảng. Kết quả là thuật toán sẽ hoạt động với độ phức tạp  $O(n^2)$  → không hiệu quả.
  - **Trường hợp 2:** Chọn phần tử trung vị '5' làm pivot và áp dụng Quick Sort, ta thấy sau lần phân hoạch thứ nhất, mảng được chia ra làm 2 phần {1,2,3,4} và {6,7,8,9,10}, làm giảm đáng kể kích thước của mảng. Quá trình phân hoạch hiệu quả hơn và thuật toán hoạt động với độ phức tạp  $O(n \log n)$ , tối ưu hơn so với trường hợp trước.
  - **Trường hợp 3:** Chọn phần tử cuối cùng làm pivot, tức là '10' và áp dụng Quick Sort. Tương tự như trường hợp 1, phần bên phải (các phần tử lớn hơn 10) là mảng rỗng → không làm giảm kích thước của mảng → độ phức tạp  $O(n^2)$  → không hiệu quả.
- c) Thuật toán sắp xếp nhanh (Quick Sort) sẽ có hiệu quả tốt nhất trong trường hợp này.
- Lý do đề xuất: Vì trong trường hợp mảng đã sắp xếp hoặc gần như đã sắp xếp (trường hợp tốt nhất), khi ta chọn pivot là phần tử đầu/cuối/trung vị, Quick Sort đều sẽ có độ phức tạp là  $O(n \log n)$ , tối ưu hơn so với các giải thuật sắp xếp khác.

## Câu 2 (4 điểm):

Cho dãy số: 11, 6, 8, 19, 4, 10, 5, 17, 43, 49, 31

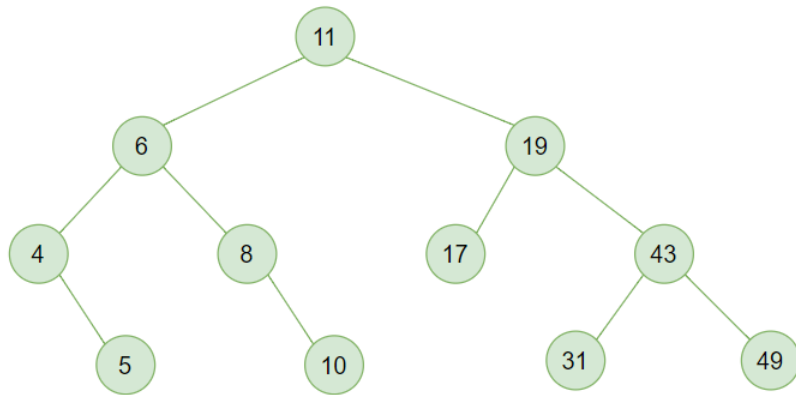
Hãy thực hiện các yêu cầu sau:

- a) Xây dựng **cây nhị phân tìm kiếm** từ dãy số đã cho bằng cách thêm (insert) vào cây các số của dãy số theo thứ tự từ trái qua phải. Mỗi nút (node) của cây chứa một số của dãy số. (1 điểm)
- b) Duyệt cây trong câu 2.a) theo Node – Left – Right, Right – Left – Node.
- c) Vẽ từng bước thực hiện xóa khỏi cây nút 11 sao cho cây vẫn là cây nhị phân tìm kiếm sau khi xóa nút. (1 điểm)
- d) Vẽ từng bước thực hiện xây dựng **cây nhị phân tìm kiếm cân bằng**, bằng cách thêm (insert) vào cây các số của dãy số theo thứ tự từ trái qua phải. Ghi rõ loại mất cân bằng và thao tác xử lý khi tái cân bằng cây. (1 điểm)

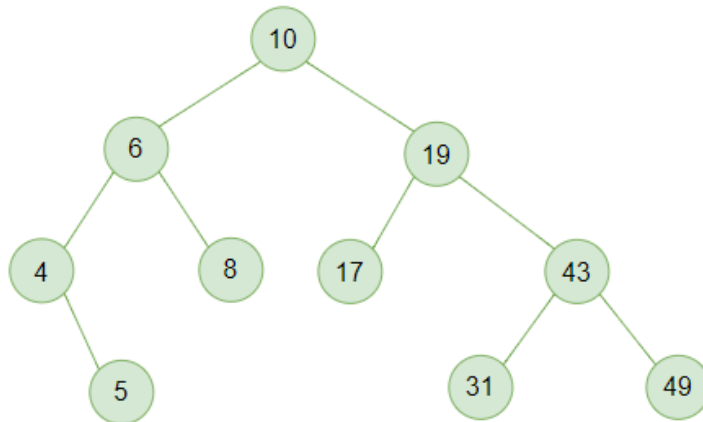


### Đáp án tham khảo:

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



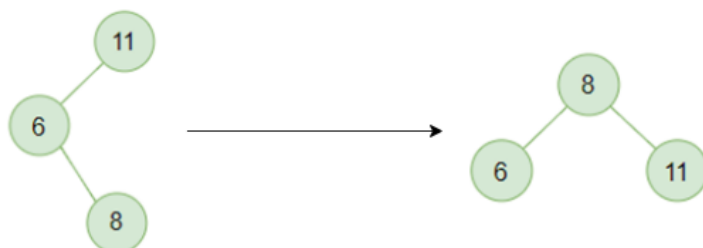
- b) Duyệt cây theo Node – Left – Right: 11, 6, 4, 5, 8, 10, 19, 17, 43, 31, 49  
 Duyệt cây theo Right – Left – Node: 43, 31, 43, 17, 19, 10, 8, 5, 4, 6, 11
- c) Nút 11 là nút có đầy đủ 2 cây con, chọn nút thế mạng có hai cách:
- Cách 1: chọn nút lớn nhất của cây con bên trái
  - Cách 2: chọn nút nhỏ nhất của cây con bên phải
- Ở đây, ta chọn cách 1 → nút thế mạng là 10.



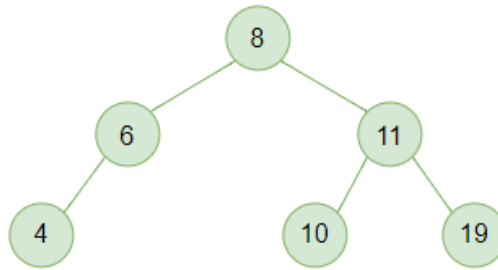
- d) Cây nhị phân tìm kiếm cân bằng là cây nhị phân mà tại mỗi nút độ cao của cây con trái và của cây con phải chênh lệch không quá một.
- Thêm 11, 6: cây chưa mất cân bằng



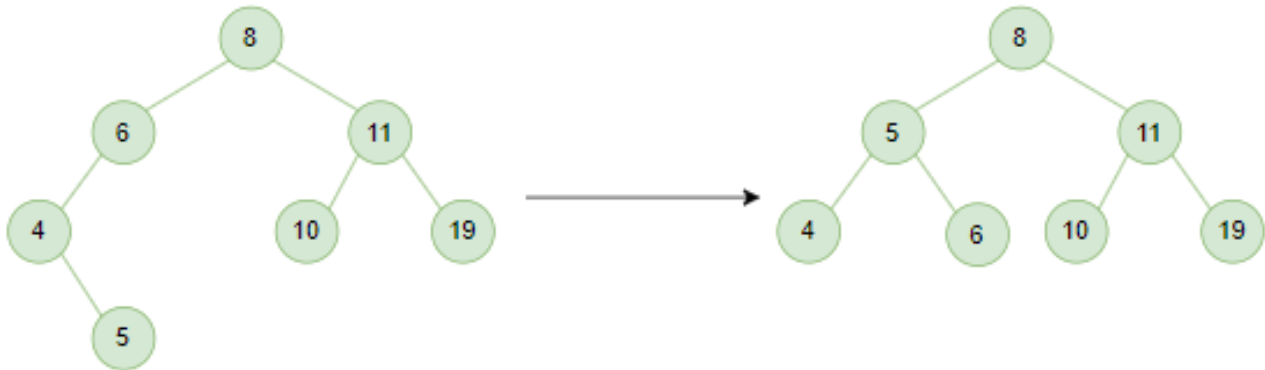
- Thêm 8: cây mất cân bằng L – R; thực hiện phép quay trái → 6 trở thành cây con của 8. Cây vẫn còn mất cân bằng L – L, thực hiện quay phải để 8 thành nút gốc mới của cây → 11 trở thành cây con bên phải của chính cây con bên trái của nó.



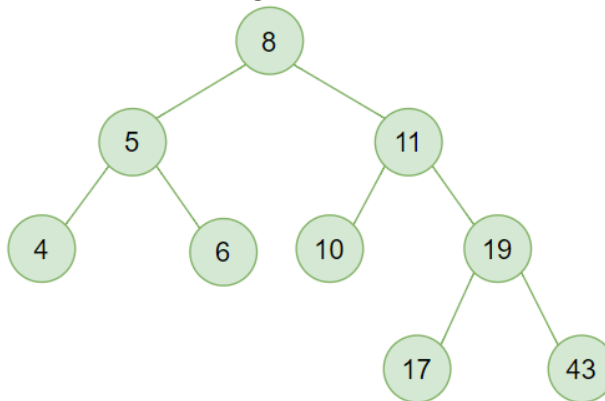
- Thêm 19, 4, 10: cây chưa mất cân bằng



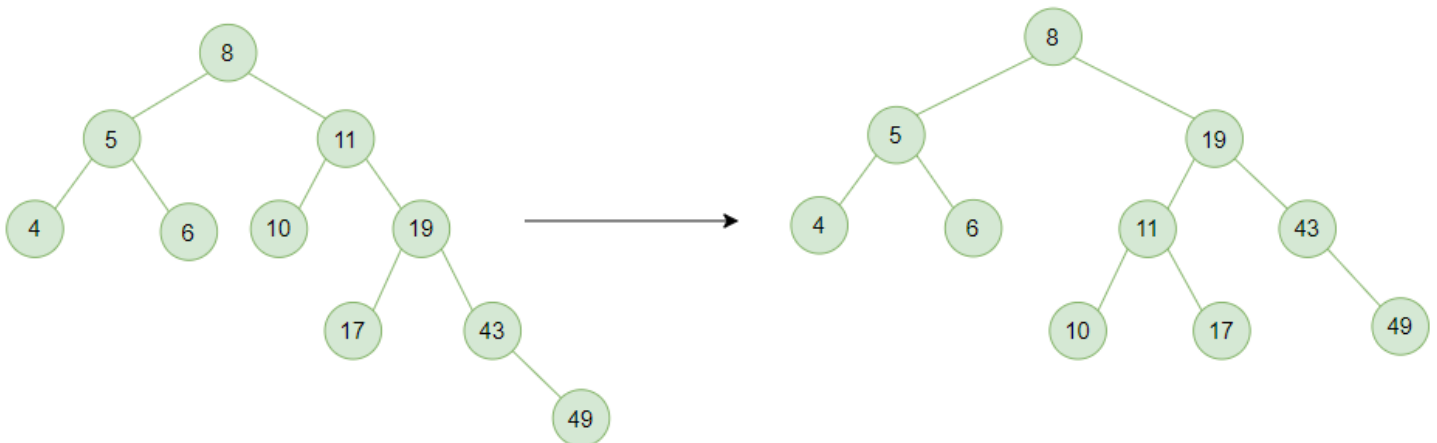
- Thêm 5: cây mất cân bằng L – R; thực hiện phép quay trái → 4 trở thành cây con của 5. Cây vẫn còn mất cân bằng L – L, thực hiện quay phải để 5 thành nút gốc mới → 6 trở thành cây con bên phải của 5.



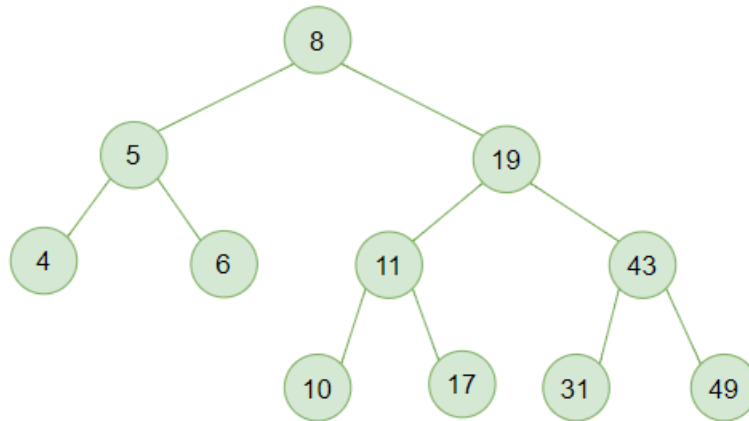
- Thêm 17, 43: cây chưa mất cân bằng



- Thêm 49: cây mất cân bằng R – L; thực hiện phép quay trái → 19 trở thành nút gốc mới của cây con bên phải của gốc 8.



➤ Thêm 31: cây không mất cân bằng, ta được cây nhị phân tìm kiếm cân bằng



**Câu 3 (3 điểm):**

Bãi xe ô tô của Trường U có 13 chỗ, đánh số từ 1 đến 13. Trong buổi sáng ngày 17 tháng 6 năm 2019, có 10 xe cần vào bãi xe. Bảng số của các xe **lần lượt** vào bãi là {10, 26, 52, 76, 13, 08, 03, 33, 60, 42}.

- Anh/chị hãy đề xuất một giải pháp dựa trên kỹ thuật hashing (băm) để đưa các xe vào bãi, sao cho chú bảo vệ có thể tìm được vị trí xe nhanh nhất dựa trên bảng số xe. (2 điểm)
- Anh/chị thực hiện từng bước đưa tất cả các xe vào bãi và sau đó tìm vị trí xe có bảng số 33 theo giải pháp đề xuất. (1 điểm)

**Đáp án tham khảo:**

- Ta sử dụng bảng băm theo phương pháp thăm dò tuyến tính:
  - Hàm băm:  $H(\text{key}) = (\text{key} \% M) + 1$
  - Hàm băm lại (khi xảy ra đụng độ):  $\text{prob}(\text{key}, i) = ((h(\text{key}) + i) \% M) + 1$
  - Trong đó:
    - key là bảng số xe
    - $H(\text{key})$  là vị trí đỗ xe trong bãi
    - $i$  là một số nguyên cho biết số lần băm lại (nếu vị trí đó đã có xe đỗ)
    - $M$  là vị trí (chỗ) trong bãi đỗ xe,  $M = 13$
- Thêm 10:  $H(10) = 10 \% 13 + 1 = 11$ . Thêm '10' vào vị trí 11.

Thêm 26:  $H(26) = 26 \% 13 + 1 = 1$ . Thêm '26' vào vị trí 1.

Thêm 52:  $H(52) = 52 \% 13 + 1 = 1$ . Vị trí 1 chứa '26' → xảy ra đụng độ.

+ Băm lại lần 1:  $H(52, 1) = (52 + 1) \% 13 + 1 = 2$ . Thêm '52' vào vị trí 2.

Thêm 76:  $H(76) = 76 \% 13 + 1 = 10$ . Thêm '76' vào vị trí 10.

Thêm 13:  $H(13) = 13 \% 13 + 1 = 1$ . Vị trí 1 chứa '26' → xảy ra đụng độ.

+ Băm lại lần 1:  $H(13, 1) = (13 + 1) \% 13 + 1 = 2$  → xảy ra đụng độ

+ Băm lại lần 2:  $H(13, 2) = (13 + 2) \% 13 + 1 = 3$ . Thêm '13' vào vị trí 3.

Thêm 8:  $H(8) = 8 \% 13 + 1 = 9$ . Thêm '8' vào vị trí 9.

Thêm 3:  $H(3) = 3 \% 13 + 1 = 4$ . Thêm '3' vào vị trí 4.

Thêm 33:  $H(33) = 33 \% 13 + 1 = 8$ . Thêm '33' vào vị trí 8

Thêm 60:  $H(60) = 60 \% 13 + 1 = 9$ . Vị trí 9 chứa '8' → xảy ra đụng độ.

+ Băm lại lần 1:  $H(60, 1) = (60 + 1) \% 13 + 1 = 10$  → xảy ra đụng độ

- + Băm lại lần 2:  $H(60,2) = (60 + 2) \% 13 + 1 = 11 \rightarrow$  xảy ra đụng độ
- + Băm lại lần 3:  $H(13,3) = (60 + 3) \% 13 + 1 = 12$ . Thêm '60' vào vị trí 12.
- Thêm 42:  $H(42) = 42 \% 13 + 1 = 4$ . Vị trí 4 chứa '3'  $\rightarrow$  xảy ra đụng độ
- + Băm lại lần 1:  $H(42,1) = (42 + 1) \% 13 + 1 = 5$ . Thêm '42' vào vị trí 5.

Bảng kết quả sau khi băm:

	<i><b>Key</b></i>
1	26
2	52
3	13
4	3
5	42
6	
7	
8	33
9	8
10	76
11	10
12	60
13	

Tìm 33:  $H(33) = 33 \% 13 + 1 = 8$ . Vị trí 8 chứa '33'  $\rightarrow$  thông báo đã tìm thấy bảng số xe 33 trong bãi

**HẾT**