

BAN HỌC TẬP KHOA CÔNG NGHỆ PHẦN MỀM

CHUỖI TRAINING CUỐI HỌC KÌ 2 NĂM HỌC 2021 - 2022



Sharing is learning



Ban học tập

Khoa Công Nghệ Phần Mềm
Trường ĐH Công Nghệ Thông Tin
ĐHQG Hồ Chí Minh



Email / Group

bht.cnpm.uit@gmail.com
fb.com/groups/bht.cnpm.uit

Training

Cấu trúc dữ liệu và Giải thuật

🕒 Thời gian training: 10h ngày 17/06/2022

👤 Trainer: Mai Hiếu Hiền - KHCL2020
Phan Phạm Trọng Phúc – ATTN2021



Sharing is learning



Sharing is learning

Nội dung Training

I. Các thuật toán sắp xếp

II. Cây nhị phân

III. Bảng băm

IV. Đồ thị

V. Giải đề



Sharing is learning



Sharing is learning

I. Các thuật toán sắp xếp

1. **Selection sort**
2. **Insertion sort**
3. **Quick sort**
4. **Merge sort**
5. **Heap sort**



Sharing is learning



Sharing is learning

I. Các thuật toán sắp xếp



Sharing is learning

1. Selection sort

Ý tưởng:

Tìm kiếm phần tử nhỏ nhất trong dãy có **N** phần tử, đưa nó về đầu dãy, tiếp tục với **N-1** phần tử còn lại. Vòng lặp kết thúc khi dãy hiện hành chỉ còn một phần tử.



Sharing is learning

I. Các thuật toán sắp xếp



Sharing is learning

1. Selection sort

Các bước tiến hành:

- Bước 1: $i=0$;
- Bước 2: Tìm phần tử $a[\min]$ nhỏ nhất trong dãy hiện hành từ $a[i]$ đến $a[n-1]$
- Bước 3: đổi chỗ $a[\min]$ và $a[i]$
- Bước 4: $i=i+1$, nếu $i < n-1$ thì lặp lại bước 2.
Ngược lại: dừng.



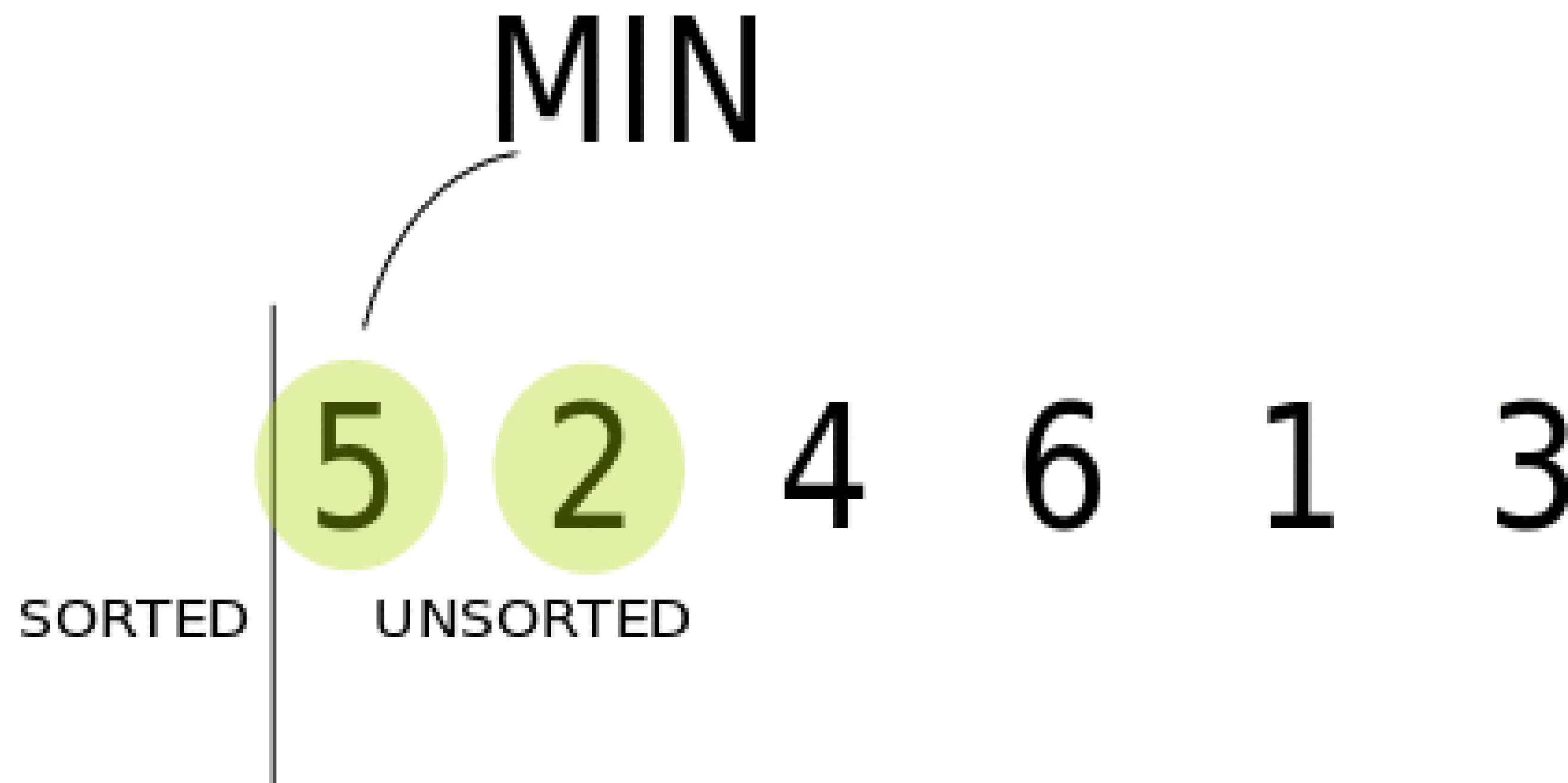
Sharing is learning

I. Các thuật toán sắp xếp



Sharing is learning

1. Selection sort



Sharing is learning

I. Các thuật toán sắp xếp



Sharing is learning

1. Selection sort

```
5
6 void selectionSort(int a[], int n)
7 {
8     int min;
9     for (int i = 0; i < n - 1; i++)
10    {
11        min = i;
12        for (int j = i + 1; j < n; j++)
13        {
14            if (a[j] < a[min])
15                min = j;
16        }
17        swap(a[min], a[i]);
18    }
19 }
20
```



Sharing is learning

I. Các thuật toán sắp xếp



Sharing is learning

1. Selection sort

Độ phức tạp :

- Best case: $O(n^2)$
- Worst case: $O(n^2)$
- Average case: $O(n^2)$



Sharing is learning

I. Các thuật toán sắp xếp

2. Insertion sort

Ý tưởng:

Giả sử ta có dãy $a[0], a[1], \dots, a[i-1]$ đã được sắp xếp, ta cần chèn $a[i]$ vào đúng vị trí của nó để $a[0] \dots a[i]$ **theo thứ tự**. Vị trí cần chèn j cần thỏa điều kiện $a[j] \leq a[i]$. Thuật toán lặp đúng $n - 1$ lần để n phần tử được sắp xếp



Sharing is learning



Sharing is learning

I. Các thuật toán sắp xếp



Sharing is learning

2. Insertion sort

Các bước tiến hành:

- Bước 1: $i=1$; //giả sử có đoạn $a[0]$ đã được sắp xếp
- Bước 2: $x=a[i]$; tìm vị trí pos thích hợp trong đoạn $a[0]$ đến $a[i-1]$ để chèn $a[i]$ vào
- Bước 3: Dời chỗ các phần tử từ $a[pos]$ đến $a[i-1]$ sang phải 1 vị trí để dành chỗ cho $a[i]$
- Bước 4: $a[pos]=x$; //có đoạn $a[1]...a[i]$ đã được sắp xếp
- Bước 5: $i=i+1$
 - ✓ Nếu $i < n$: lặp lại bước 2
 - ✓ Ngược lại: dừng



Sharing is learning

I. Các thuật toán sắp xếp

50 17 15 9 62 86 12 14 32 6

Kết quả:

- 62 50 17 15 9 86 12 14 32 6
- 86 62 50 17 15 9 12 14 32 6
- 86 62 50 17 15 12 9 14 32 6
- 86 62 50 17 15 14 12 9 32 6
- 86 62 50 32 17 15 14 12 9 6



Sharing is learning

I. Các thuật toán sắp xếp



Sharing is learning

2. Insertion sort

```
20
21 void insertionSort(int a[], int n)
22 {
23     int i, key, j;
24     for (i = 1; i < n; i++)
25     {
26         key = a[i];
27         j = i - 1;
28         while (j >= 0 && a[j] > key)
29         {
30             a[j + 1] = a[j];
31             j = j - 1;
32         }
33         a[j + 1] = key;
34     }
35 }
36
```



Sharing is learning

I. Các thuật toán sắp xếp

2. Insertion sort

Độ phức tạp :

- Best case: **$O(n)$**
- Worst case: **$O(n^2)$**
- Average case: **$O(n^2)$**



Sharing is learning



Sharing is learning

I. Các thuật toán sắp xếp



Sharing is learning

3. Quick sort

Ý tưởng:

Dựa trên thuật toán **chia để trị** (Divide and Conquer), giải thuật Quick Sort phân hoạch dãy a_1, a_2, \dots, a_n thành 2 phần:

- Phần 1: Gồm các phần tử có **giá trị bé hơn hoặc bằng** pivot
- Phần 2: Gồm các phần tử có **giá trị lớn hơn hoặc bằng** pivot

Khi đó ta sẽ có 2 mảng con: mảng bên trái của x và mảng bên phải của x. Tiếp tục công việc với mỗi mảng con (chọn pivot, phân đoạn) cho tới khi mảng được sắp xếp.



Sharing is learning

I. Các thuật toán sắp xếp



Sharing is learning

3. Quick sort

Ví dụ: thực hiện chạy từng bước giải thuật Quick sort để sắp xếp dãy số sau đây tăng dần, với pivot là phần tử đầu tiên của mảng:

16 21 45 8 11 53 3 26 49 31 12



Sharing is learning

I. Các thuật toán sắp xếp



Sharing is learning

3. Quick sort

16 21 45 8 11 53 3 26 49 31 12

16 21 45 8 11 53 3 26 49 31 12

16 12 45 8 11 53 3 26 49 31 21

16 12 45 8 11 53 3 26 49 31 21

16 12 3 8 11 53 45 26 49 31 21

16 12 3 8 11 53 45 26 49 31 21

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



Sharing is learning

I. Các thuật toán sắp xếp

3. Quick sort

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



Sharing is learning



Sharing is learning

I. Các thuật toán sắp xếp



Sharing is learning

3. Quick sort

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



Sharing is learning

I. Các thuật toán sắp xếp



Sharing is learning

3. Quick sort

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



Sharing is learning

I. Các thuật toán sắp xếp



Sharing is learning

3. Quick sort

3 8 11 12 16 21 45 26 49 31 53

3 8 11 12 16 21 45 26 49 31 53

3 8 11 12 16 21 45 26 31 49 53

3 8 11 12 16 21 45 26 31 49 53

3 8 11 12 16 21 31 26 45 49 53



Sharing is learning

I. Các thuật toán sắp xếp



Sharing is learning

3. Quick sort

3 8 11 12 16 21 31 26 45 49 53

3 8 11 12 16 21 26 31 45 49 53

3 8 11 12 16 21 26 31 45 49 53

3 8 11 12 16 21 26 31 45 49 53

3 8 11 12 16 21 26 31 45 49 53



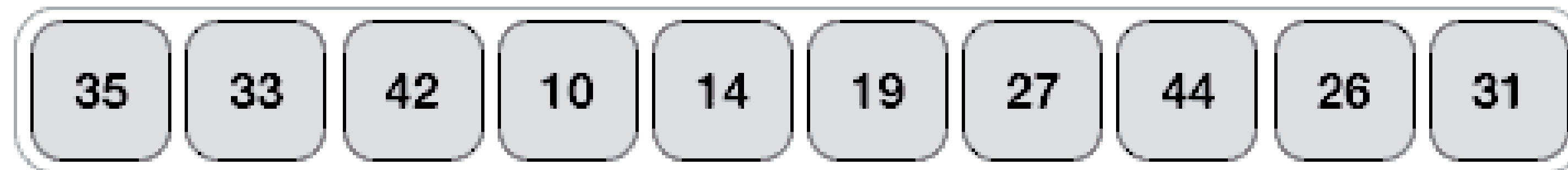
Sharing is learning

I. Các thuật toán sắp xếp

3. Quick sort

Tham khảo: thuật toán phân đoạn với pivot là phần tử cuối mảng

Unsorted Array



Sharing is learning



Sharing is learning

I. Các thuật toán sắp xếp



Sharing is learning

3. Quick sort

```
1 int partition(int a[], int l, int r){
2     int pivot = a[r];
3     int vt = l - 1;
4     for(int j = l; j < r; j++){
5         if(a[j] < pivot){
6             ++vt;
7             swap(a[vt], a[j]);
8         }
9     }
10    vt+=1;
11    swap(a[vt], a[r]);
12    return vt;
13 }
14
```

```
55
56 void quickSort(int a[], int l, int r)
57 {
58     if (l < r) {
59         int k = partition(a, l, r);
60         quickSort(a, l, k - 1);
61         quickSort(a, k + 1, r);
62     }
63 }
```

BAN HỌC TẬP

Sharing is learning

I. Các thuật toán sắp xếp

3. Quick sort

Độ phức tạp :

- Best case: **$O(n \log n)$**
- Worst case: **$O(n^2)$**
- Average case: **$O(n \log n)$**



Sharing is learning



Sharing is learning

I. Các thuật toán sắp xếp



Sharing is learning

4. Merge sort

Ý tưởng:

Dựa theo chiến thuật **Divide and Conquer**, chia dãy số thành 2 phần và tiếp tục cho đến khi chỉ còn 1 phần tử. Sau đó sắp xếp từng phần lại với nhau.



Sharing is learning

I. Các thuật toán sắp xếp

4. Merge sort

6 5 3 1 8 7 2 4



Sharing is learning



Sharing is learning

I. Các thuật toán sắp xếp

4. Merge sort

Độ phức tạp :

- Best case: **$O(n \log n)$**
- Worst case: **$O(n \log n)$**
- Average case: **$O(n \log n)$**



Sharing is learning



Sharing is learning

I. Các thuật toán sắp xếp

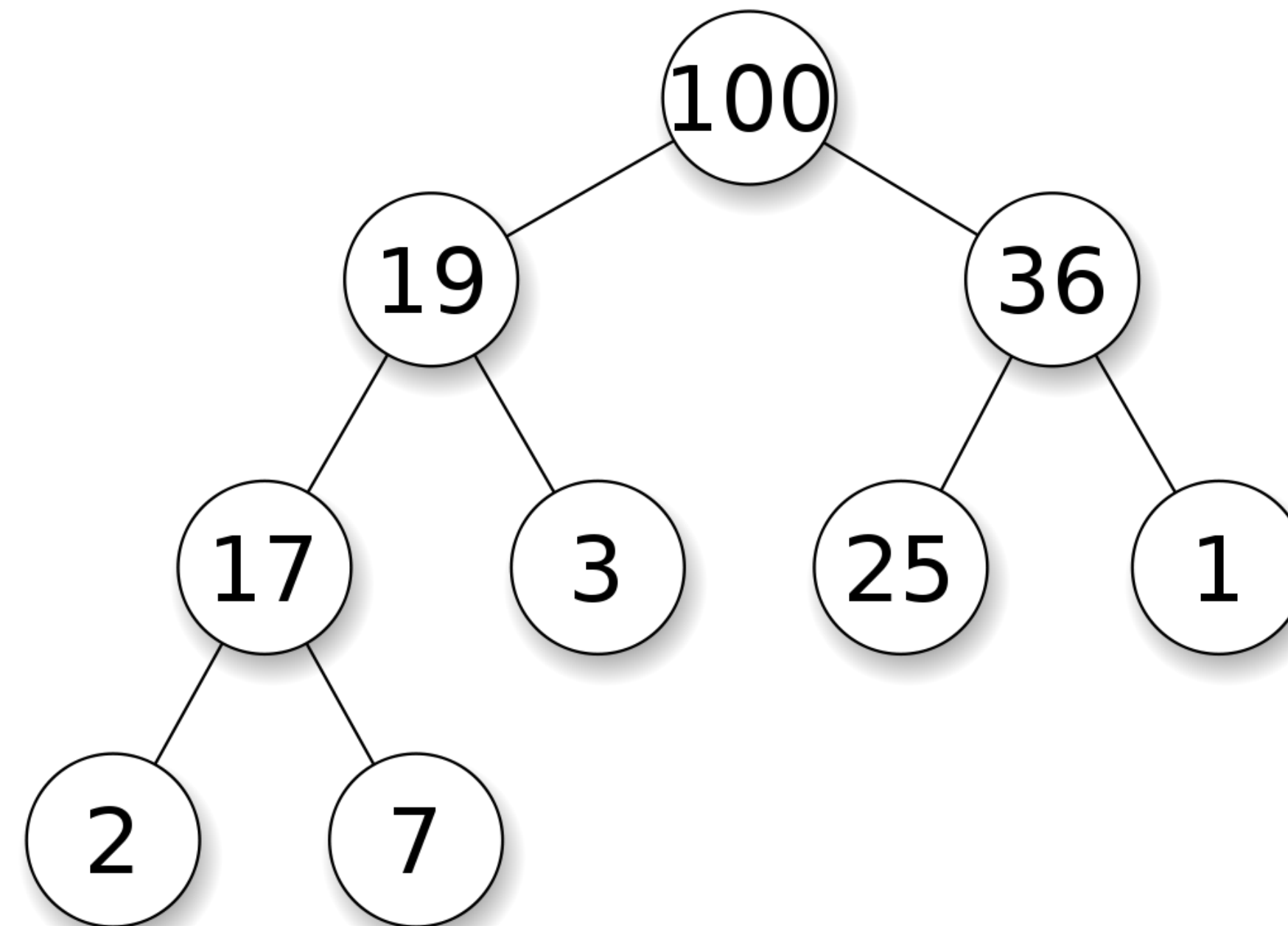


Sharing is learning

5. Heap sort

Định nghĩa heap:

- Heap là một cây nhị phân
- Giá trị của mỗi nút không bao giờ bé hơn giá trị của các nút con



Sharing is learning

I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

Ý tưởng:

Chuyển dãy cần sắp xếp thành cấu trúc heap

- ✓ Đặc điểm: tại nút thứ i tương ứng với phần tử thứ i trong mảng sẽ có **con trái là $2*i+1$** và **con phải là $2*i+2$** (nếu $2*i+1$ và $2*i+2 < n$)

Lần lượt chuyển nút gốc của heap đưa về vị trí thích hợp (về phía cuối)



Sharing is learning

I. Các thuật toán sắp xếp

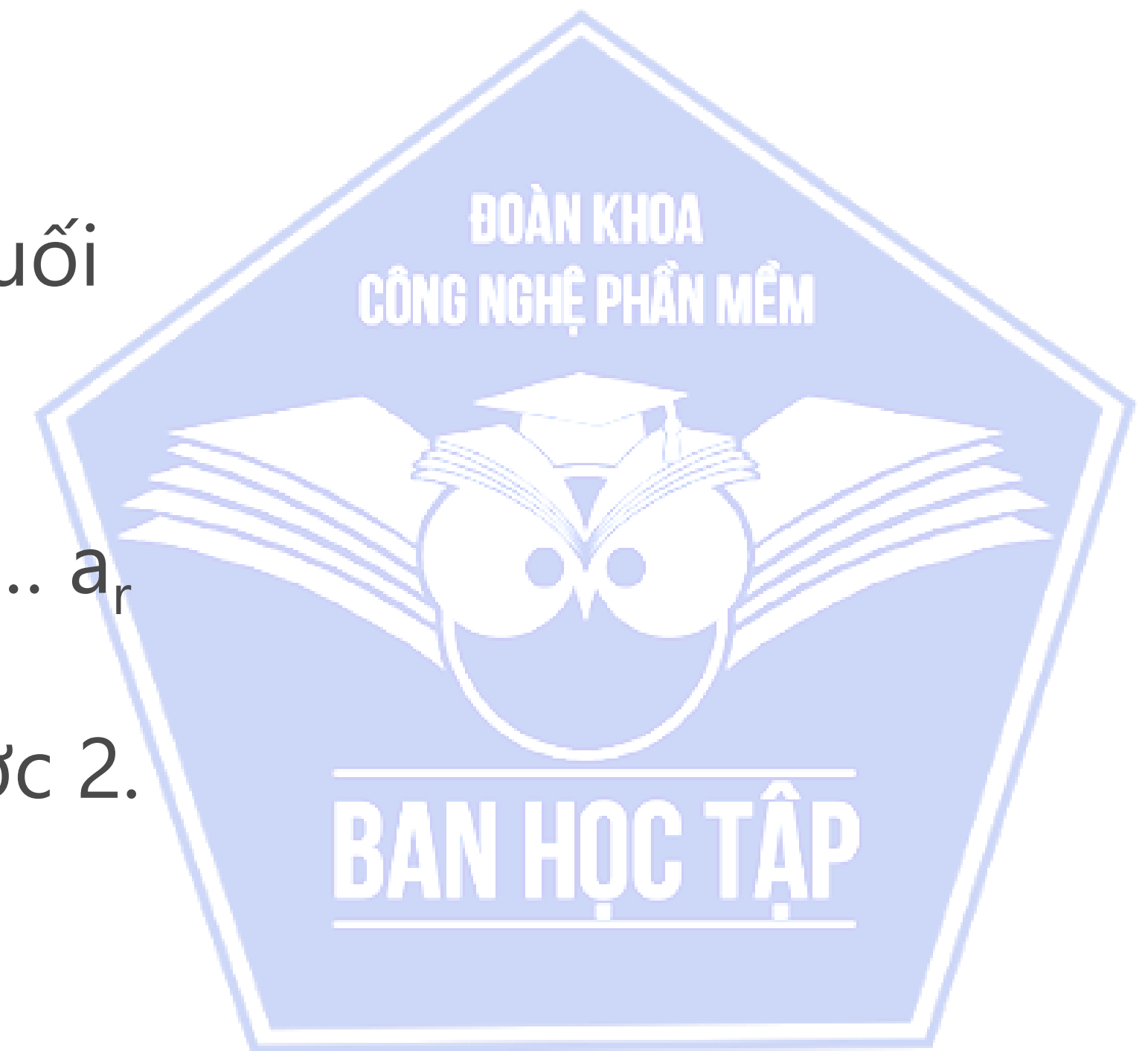


Sharing is learning

5. Heap sort

Các bước thuật toán:

- **Giai đoạn 1:** Hiệu chỉnh dãy số ban đầu thành heap
- **Giai đoạn 2:** Sắp xếp dãy số dựa trên heap
 - ✓ Bước 1: Đưa phần tử lớn nhất về vị trí đúng ở cuối dãy: $r=n$; hoán vị (a_1, a_r) ;
 - ✓ Bước 2: Loại bỏ phần tử lớn nhất ra khỏi heap: $r=r-1$; hiệu chỉnh phần còn lại của dãy từ a_1, a_2, \dots, a_r thành một heap.
 - ✓ Bước 3: Nếu $r > 1$ (heap còn phần tử), lặp lại bước 2. Ngược lại, dừng.



Sharing is learning

I. Các thuật toán sắp xếp

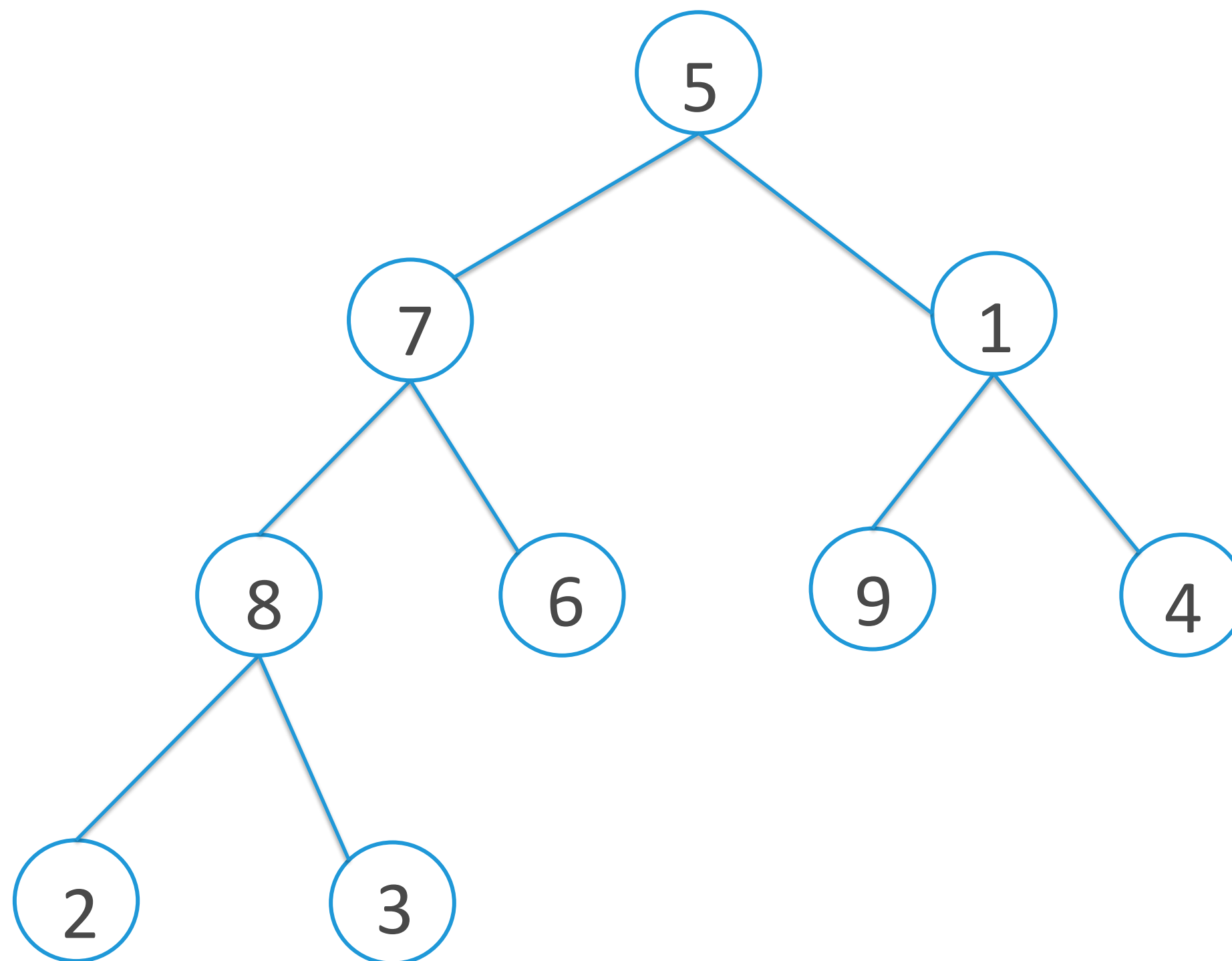


Sharing is learning

5. Heap sort:

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 5 | 7 | 1 | 8 | 6 | 9 | 4 | 2 | 3 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Giai đoạn 1: xây dựng heap (ở đây dùng max heap)



Sharing is learning

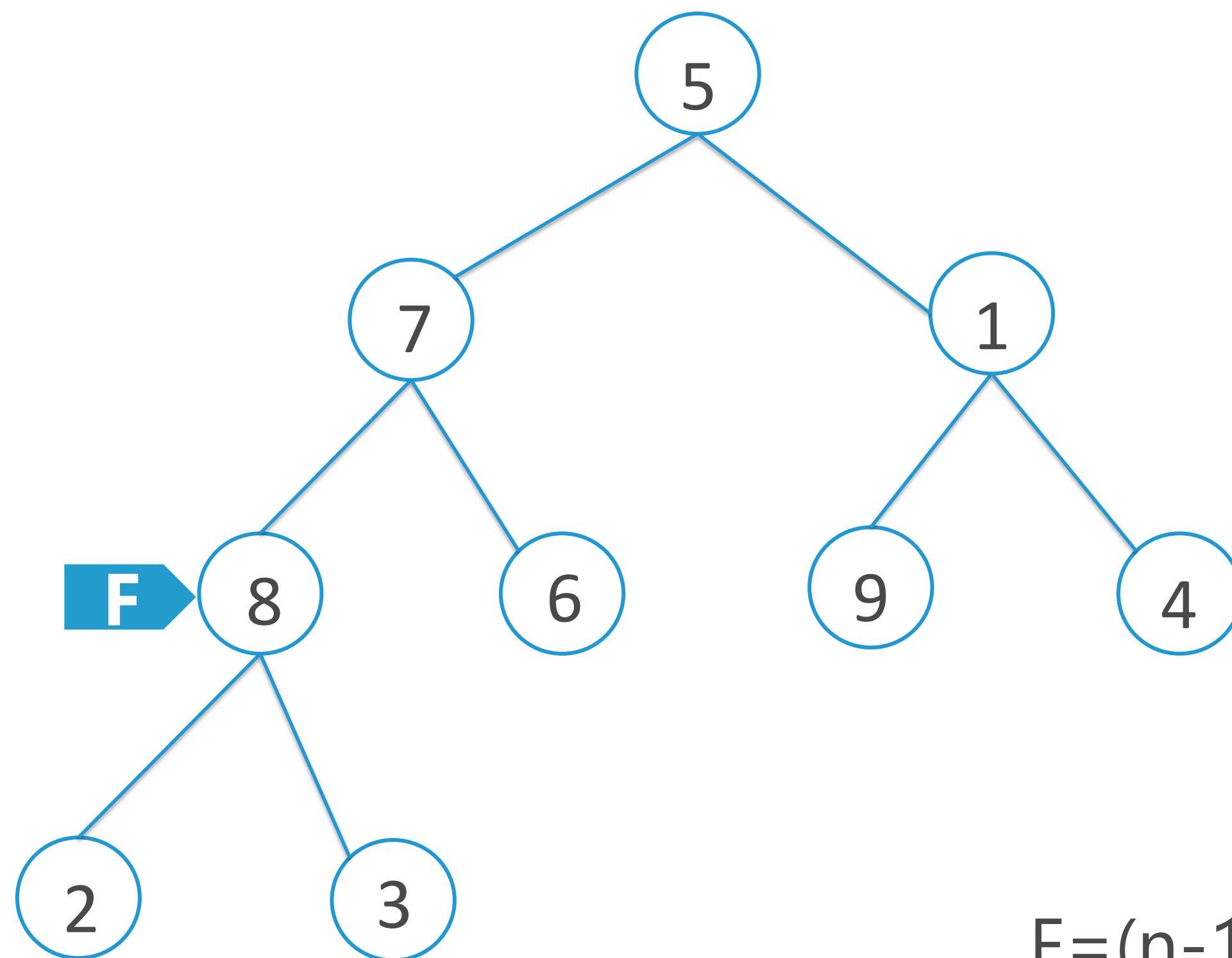
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort:

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 5 | 7 | 1 | 8 | 6 | 9 | 4 | 2 | 3 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



$$F = (n-1)/2$$



Sharing is learning

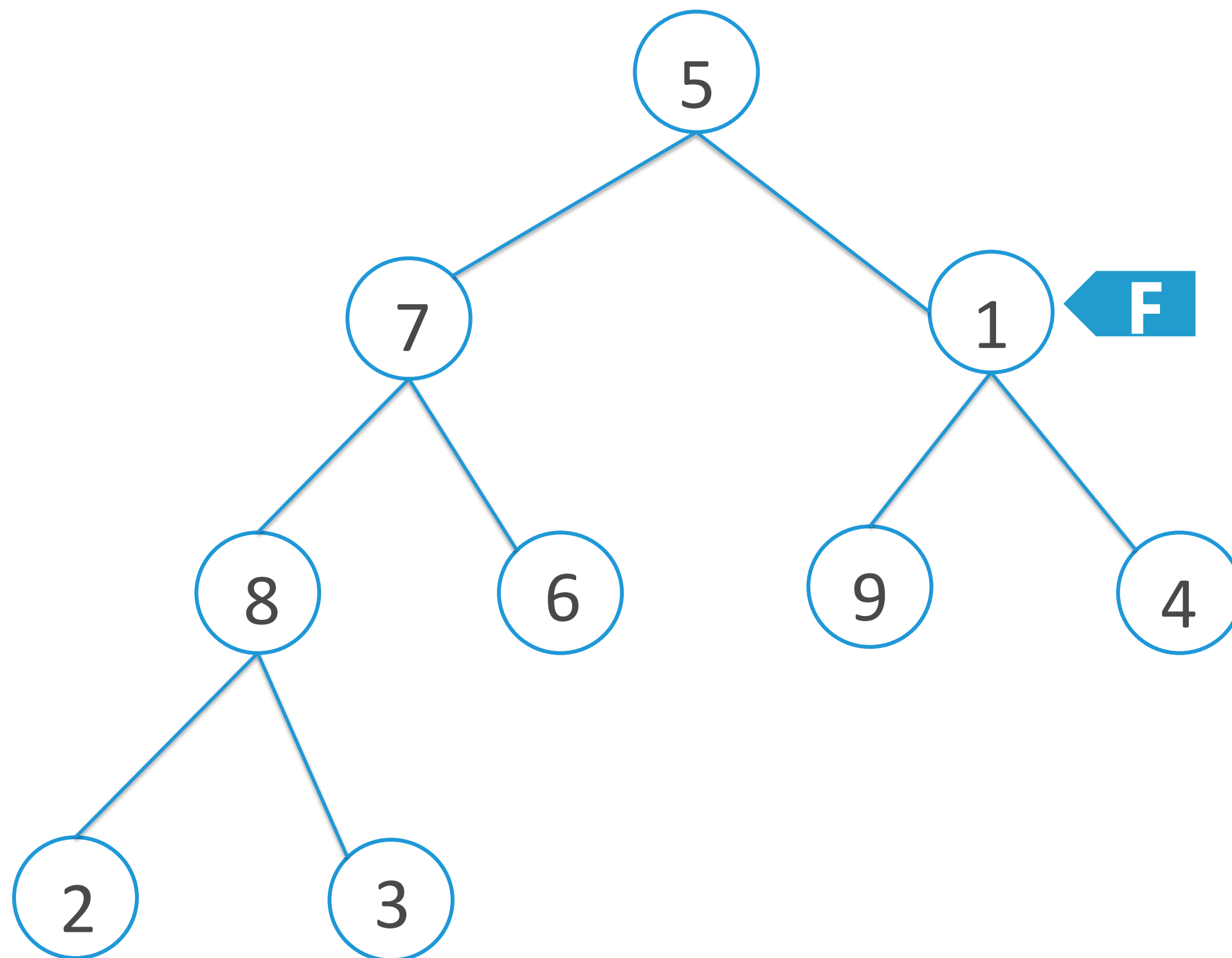
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 5 | 7 | 1 | 8 | 6 | 9 | 4 | 2 | 3 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Sharing is learning

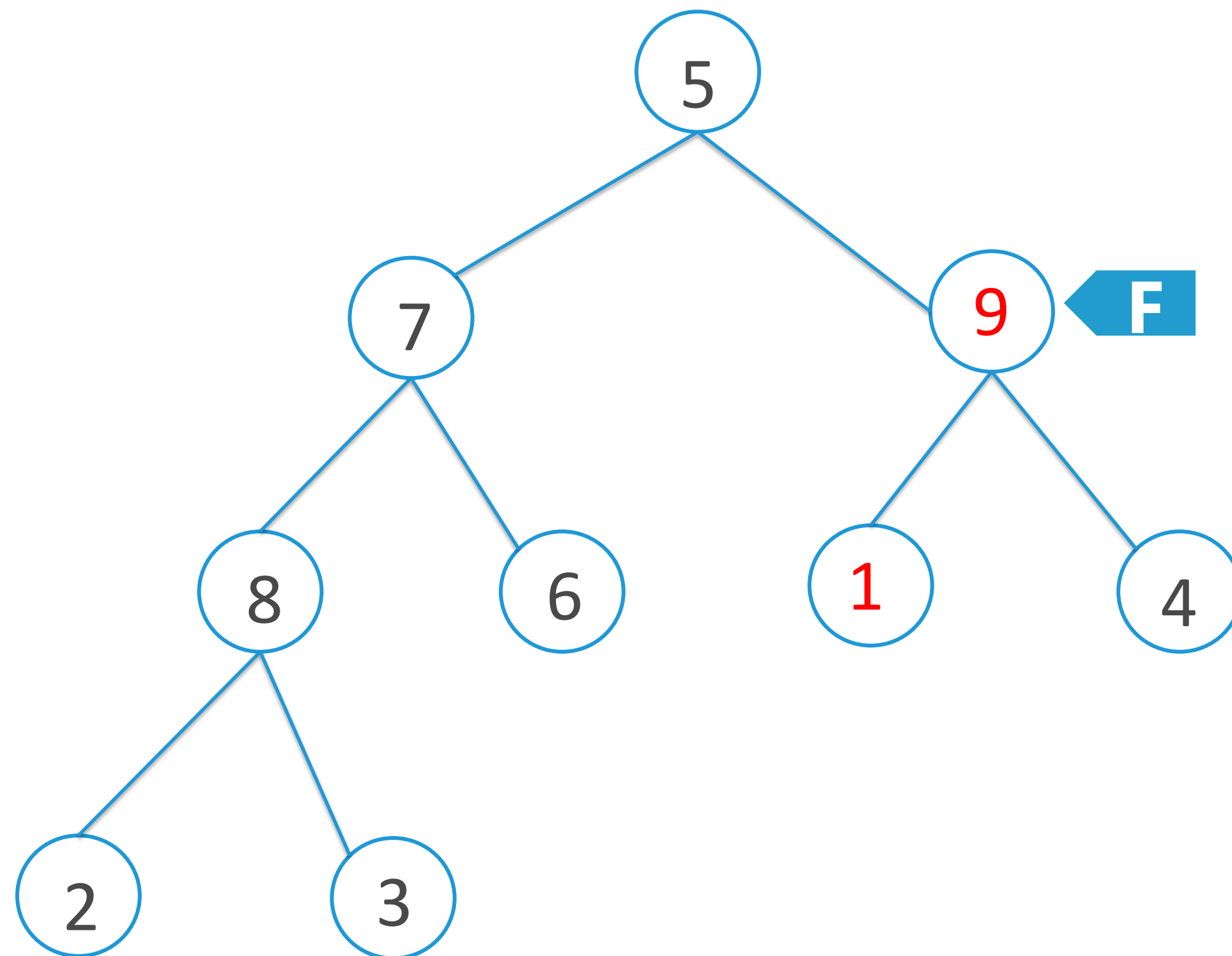
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 5 | 7 | 9 | 8 | 6 | 1 | 4 | 2 | 3 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Sharing is learning

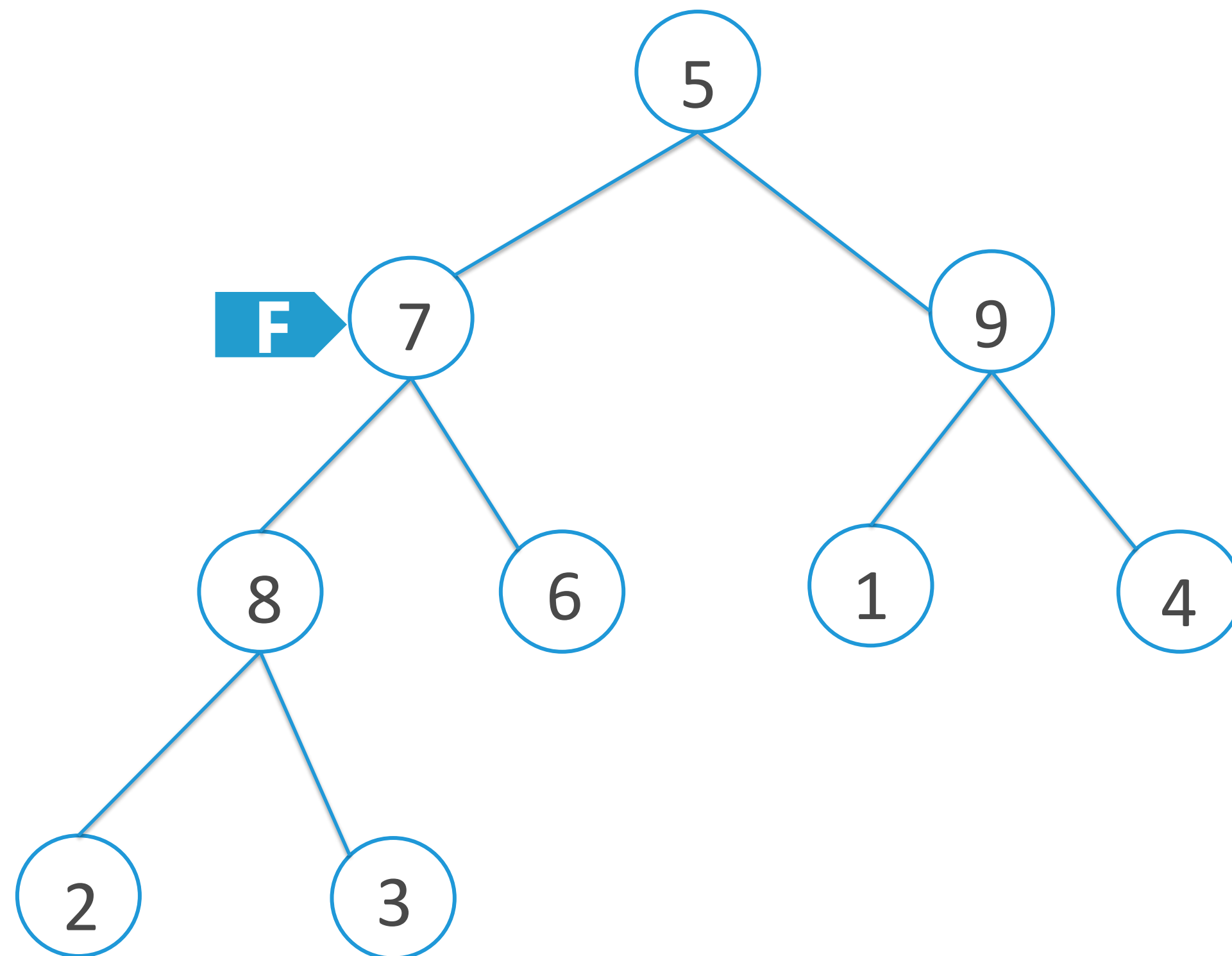
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 5 | 7 | 9 | 8 | 6 | 1 | 4 | 2 | 3 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Sharing is learning

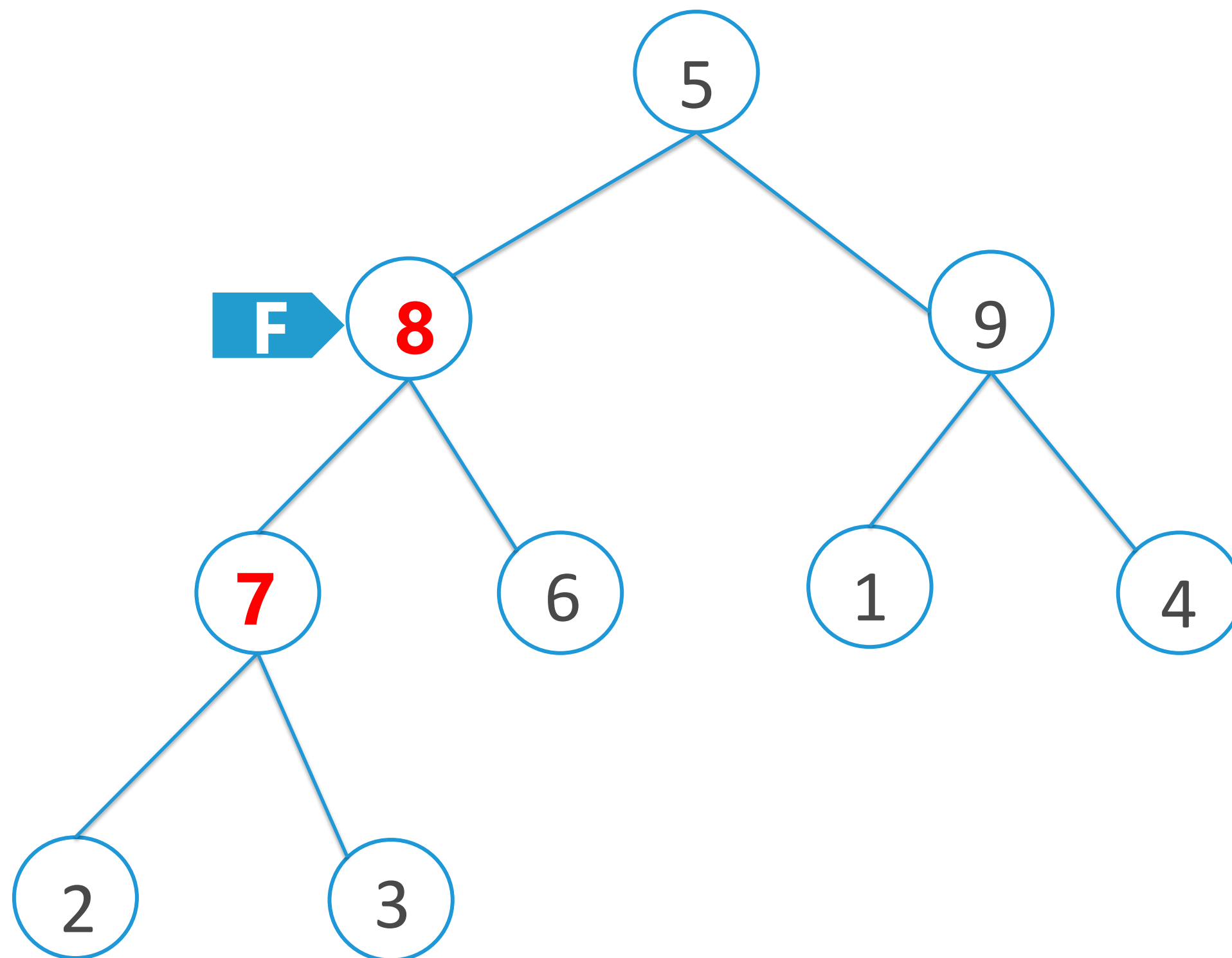
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 5 | 8 | 9 | 7 | 6 | 1 | 4 | 2 | 3 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Sharing is learning

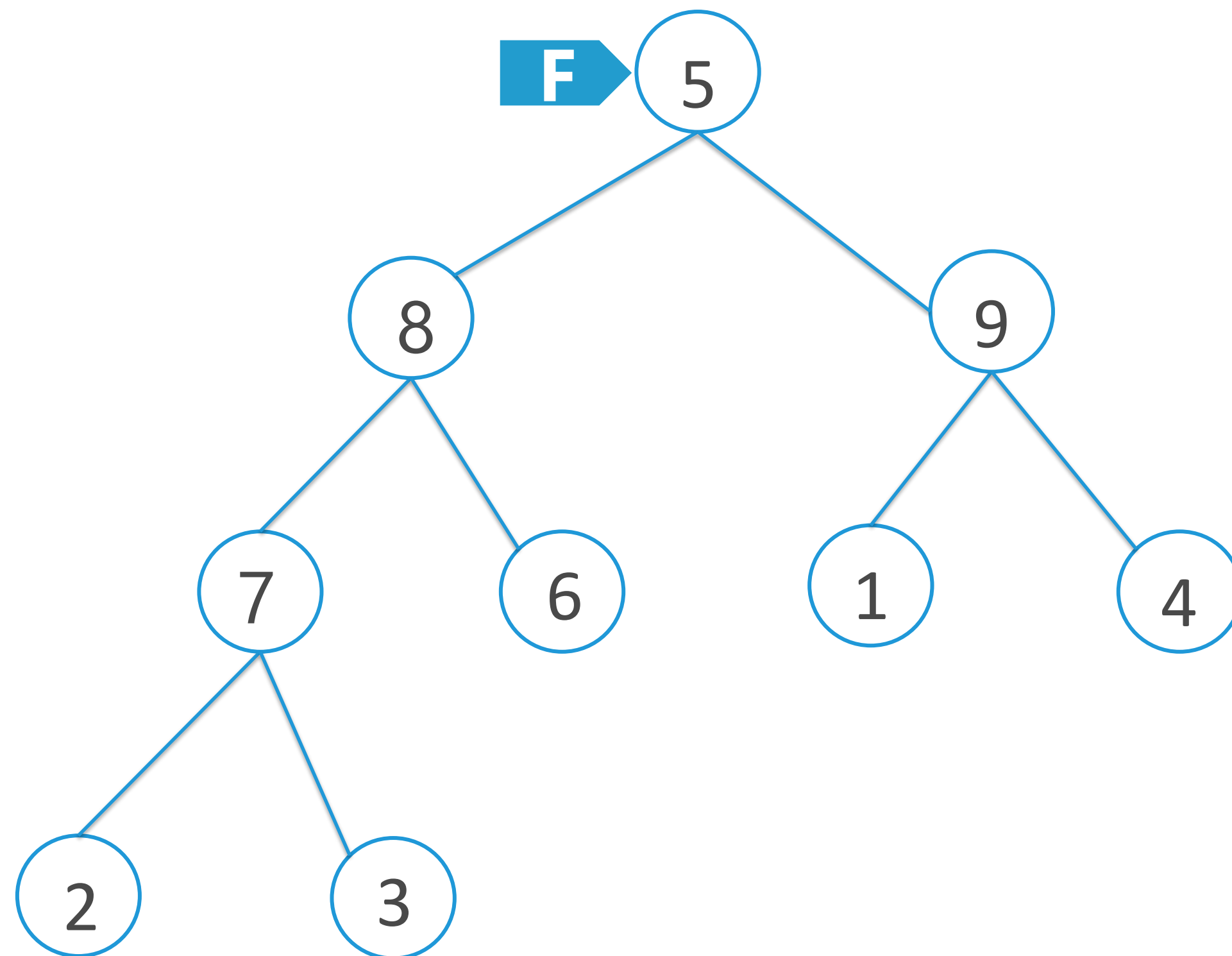
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 5 | 8 | 9 | 7 | 6 | 1 | 4 | 2 | 3 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Sharing is learning

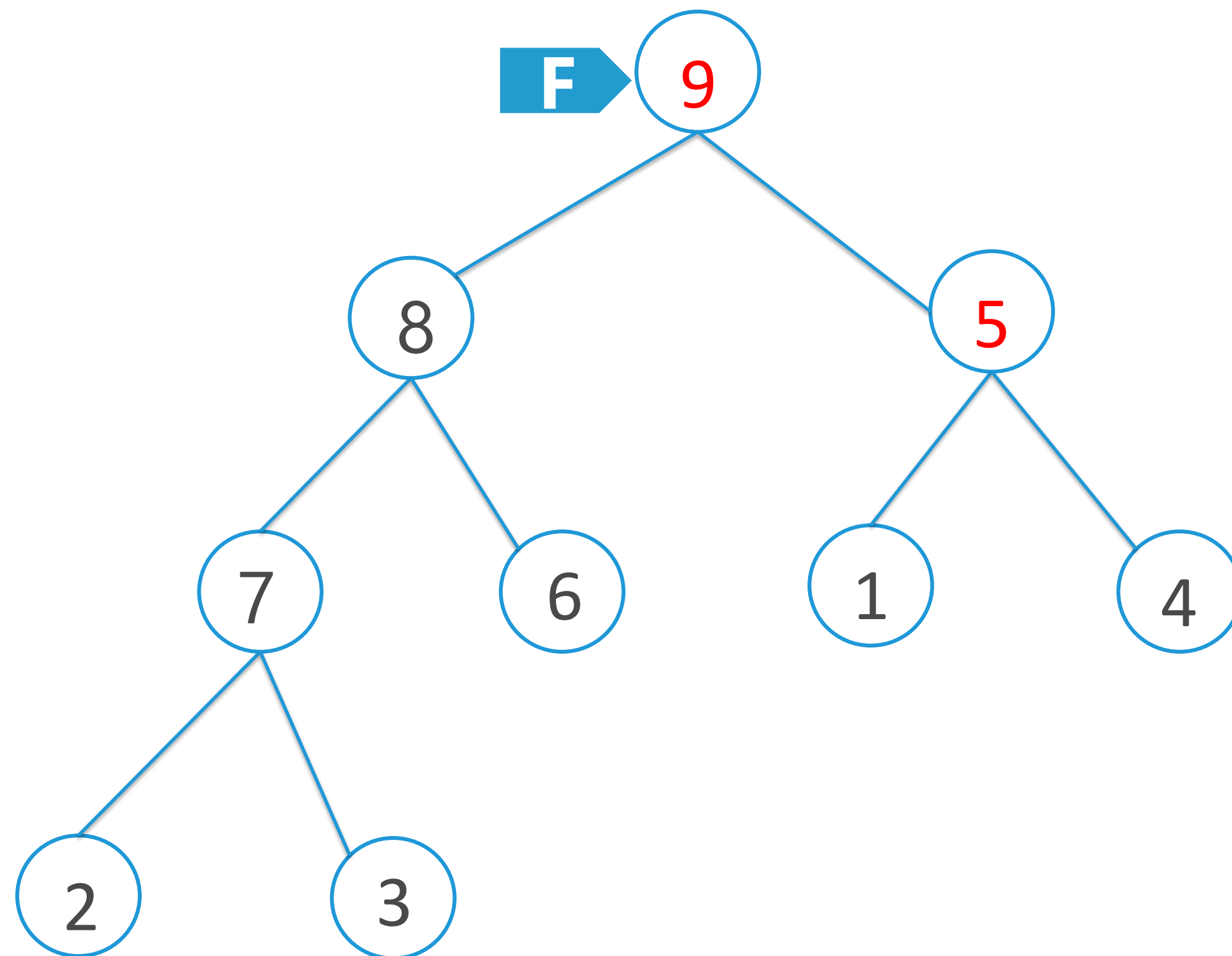
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 9 | 8 | 5 | 7 | 6 | 1 | 4 | 2 | 3 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Sharing is learning

I. Các thuật toán sắp xếp

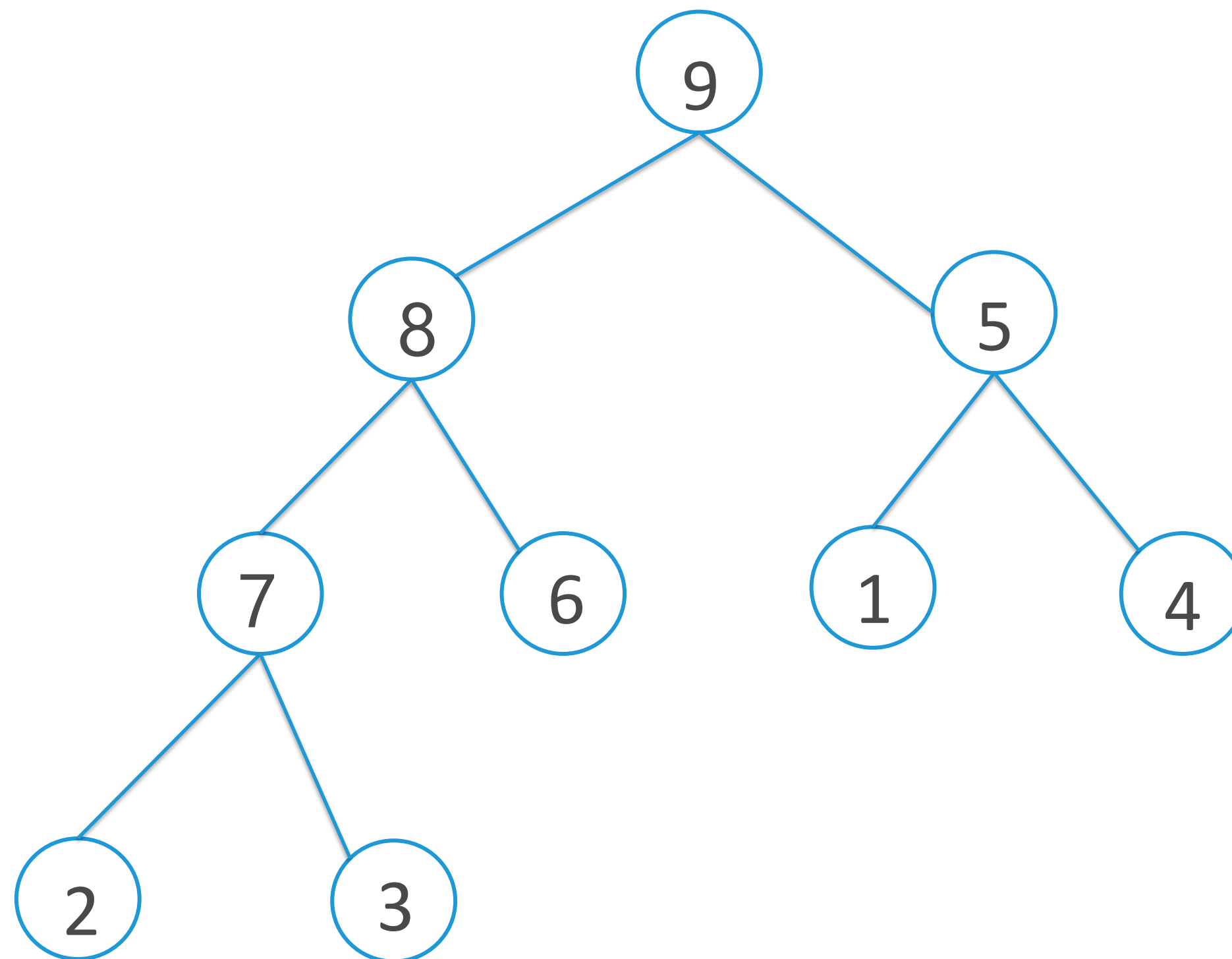


Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 9 | 8 | 5 | 7 | 6 | 1 | 4 | 2 | 3 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Giai đoạn 2: sắp xếp dãy số dựa trên heap



Sharing is learning

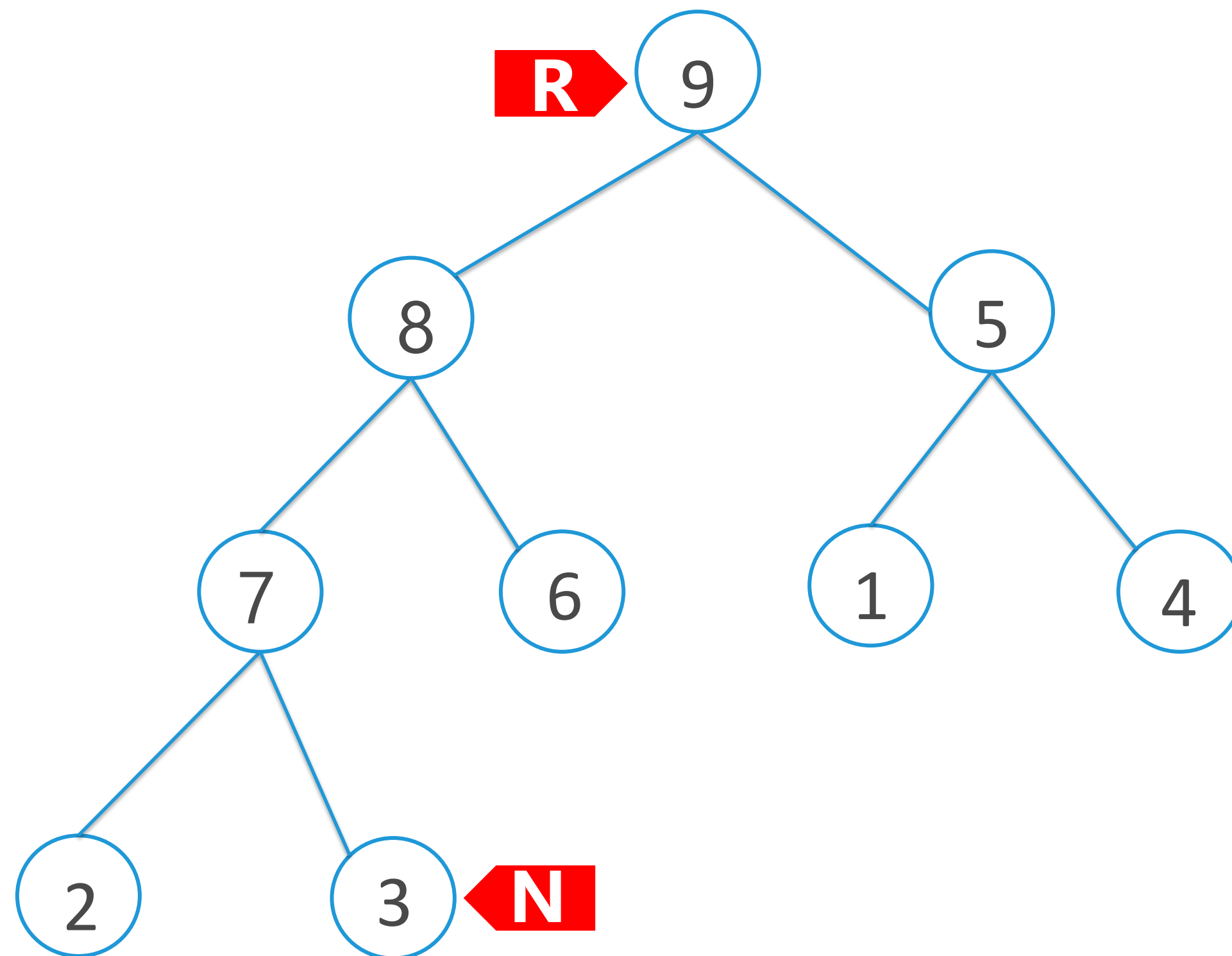
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 9 | 8 | 5 | 7 | 6 | 1 | 4 | 2 | 3 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Sharing is learning

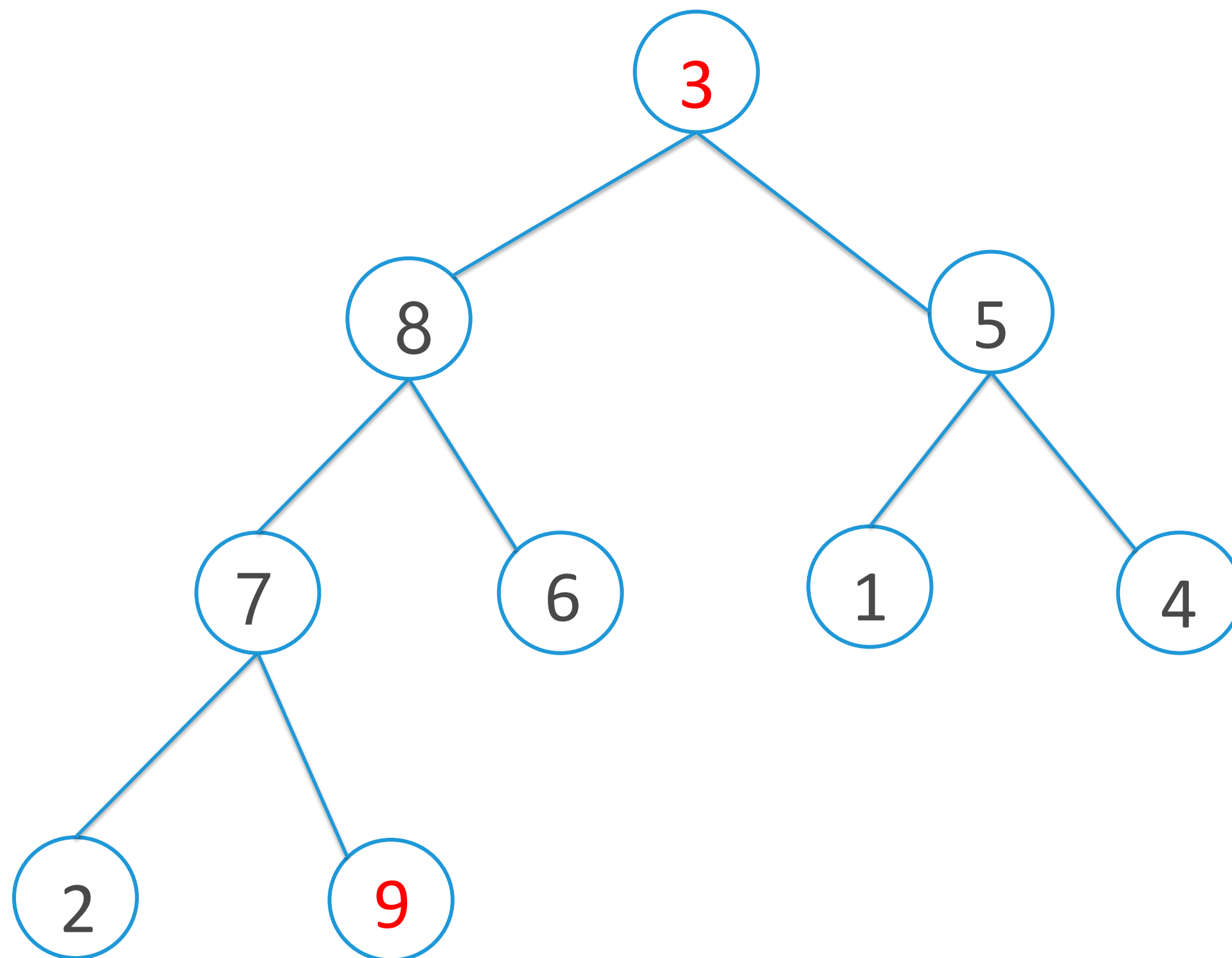
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 3 | 8 | 5 | 7 | 6 | 1 | 4 | 2 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Sharing is learning

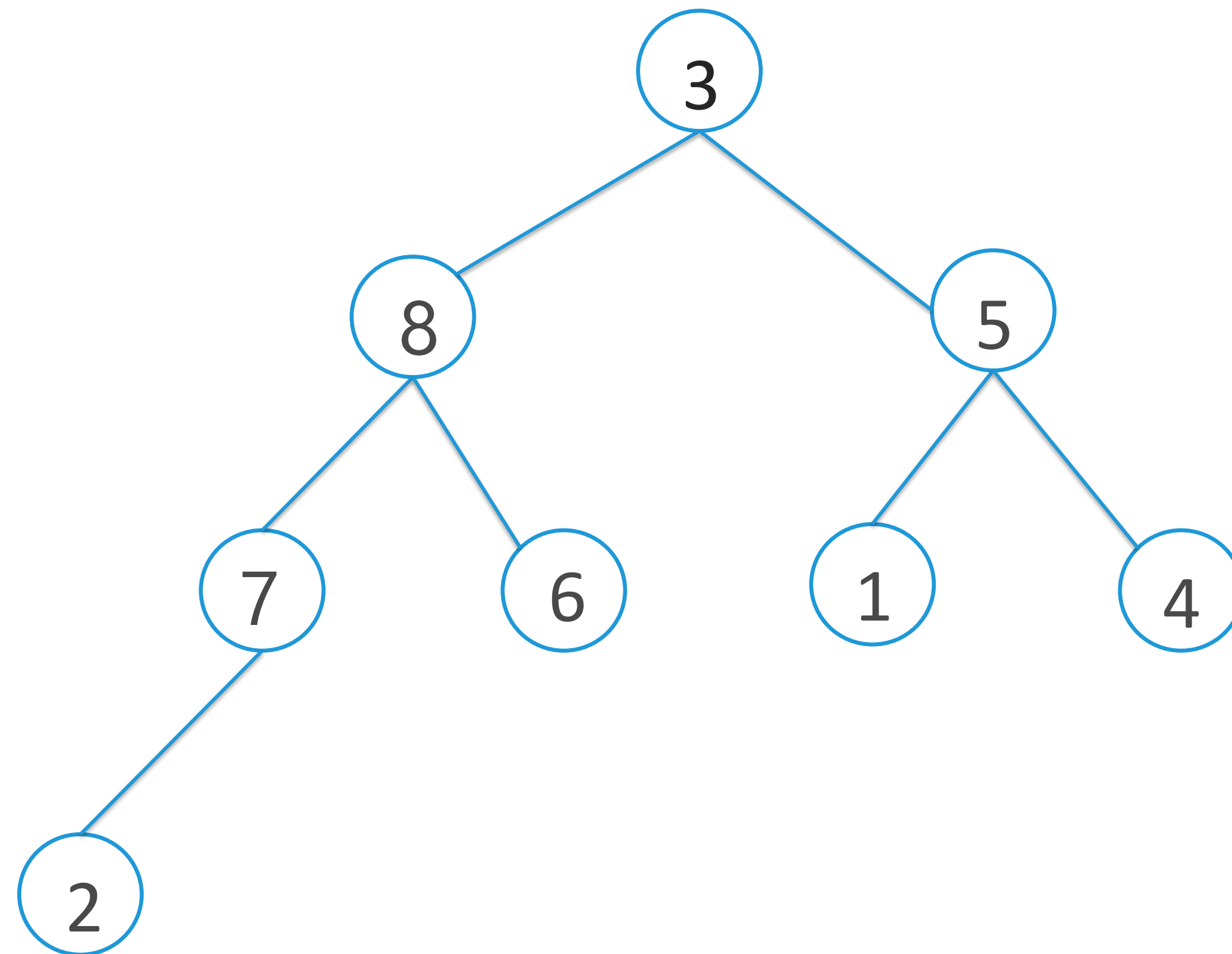
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 3 | 8 | 5 | 7 | 6 | 1 | 4 | 2 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Số phần tử còn lại: $8 > 1$, tiếp tục hiệu chỉnh dãy còn lại thành heap

Sharing is learning

I. Các thuật toán sắp xếp

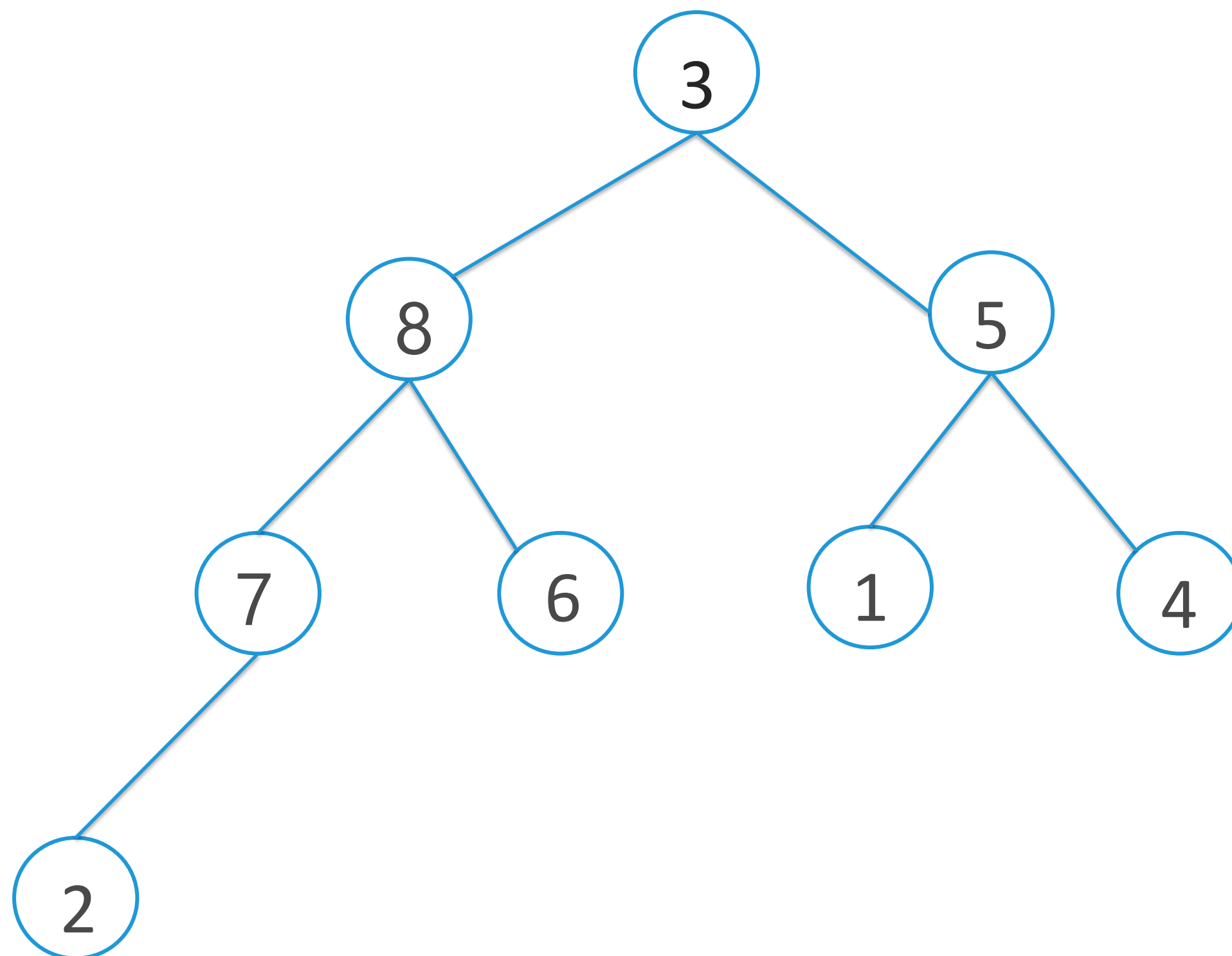


Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 3 | 8 | 5 | 7 | 6 | 1 | 4 | 2 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Tiếp tục hiệu chỉnh phần còn lại của dãy thành một heap:



Sharing is learning

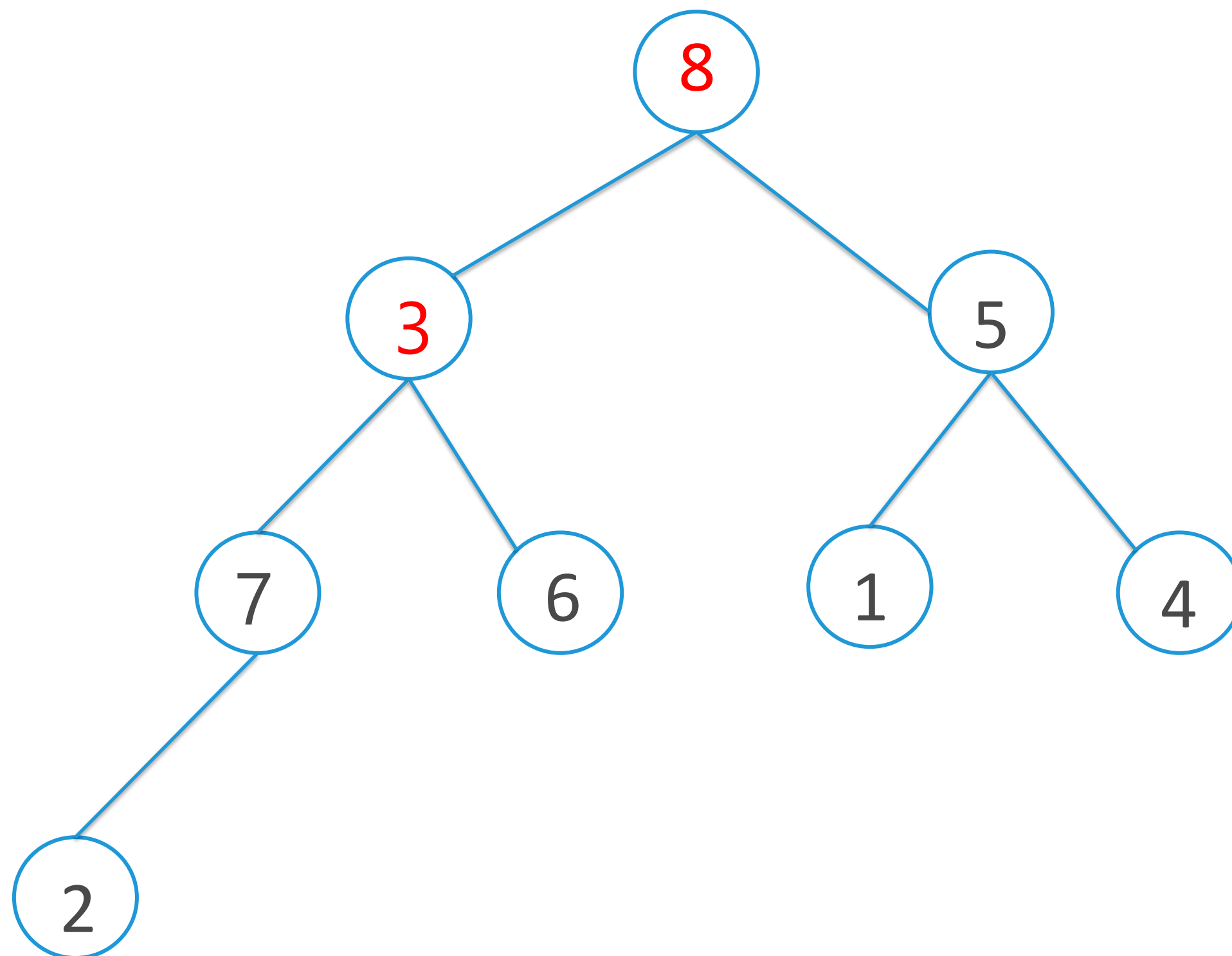
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 8 | 3 | 5 | 7 | 6 | 1 | 4 | 2 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Sharing is learning

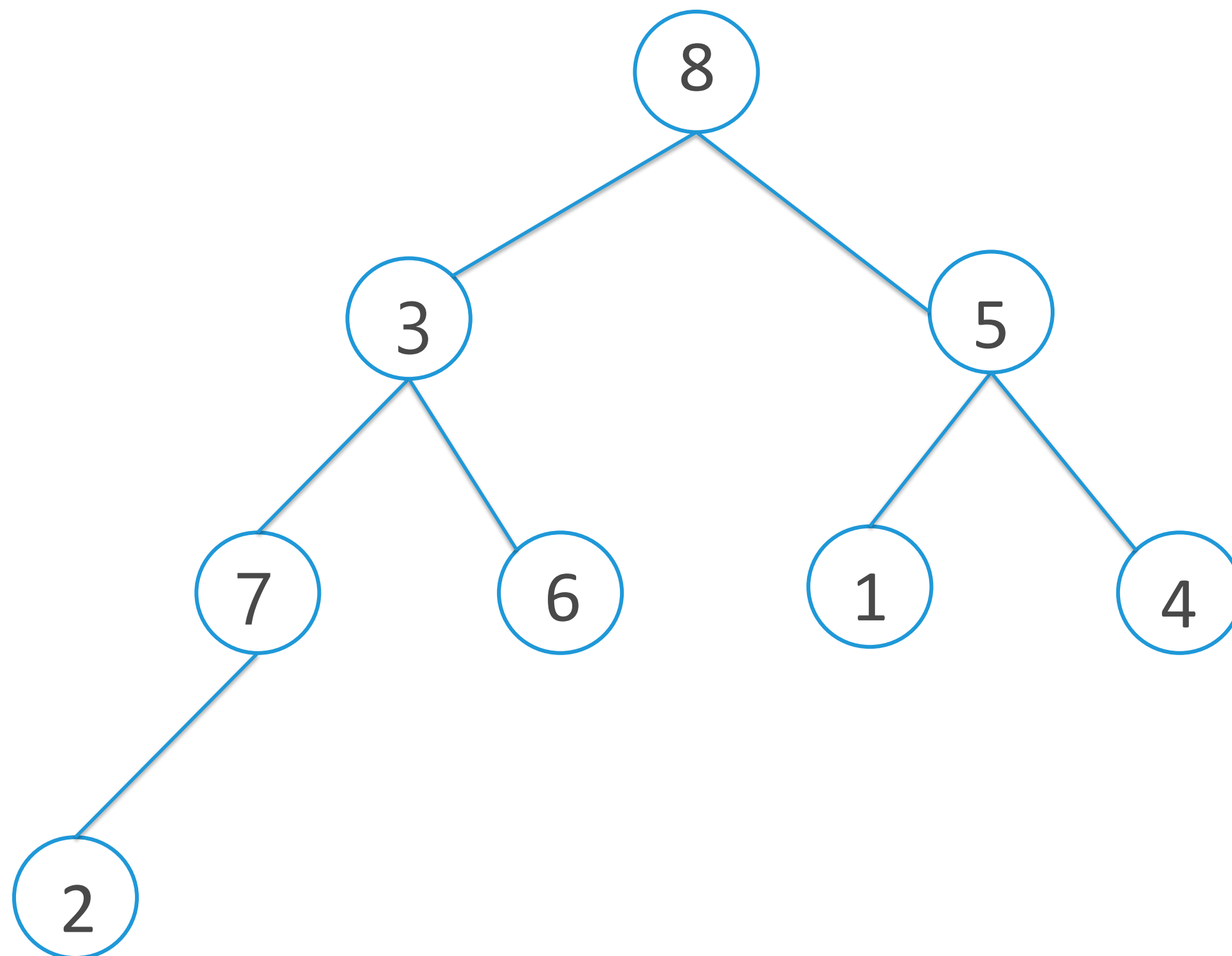
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 8 | 3 | 5 | 7 | 6 | 1 | 4 | 2 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Sharing is learning

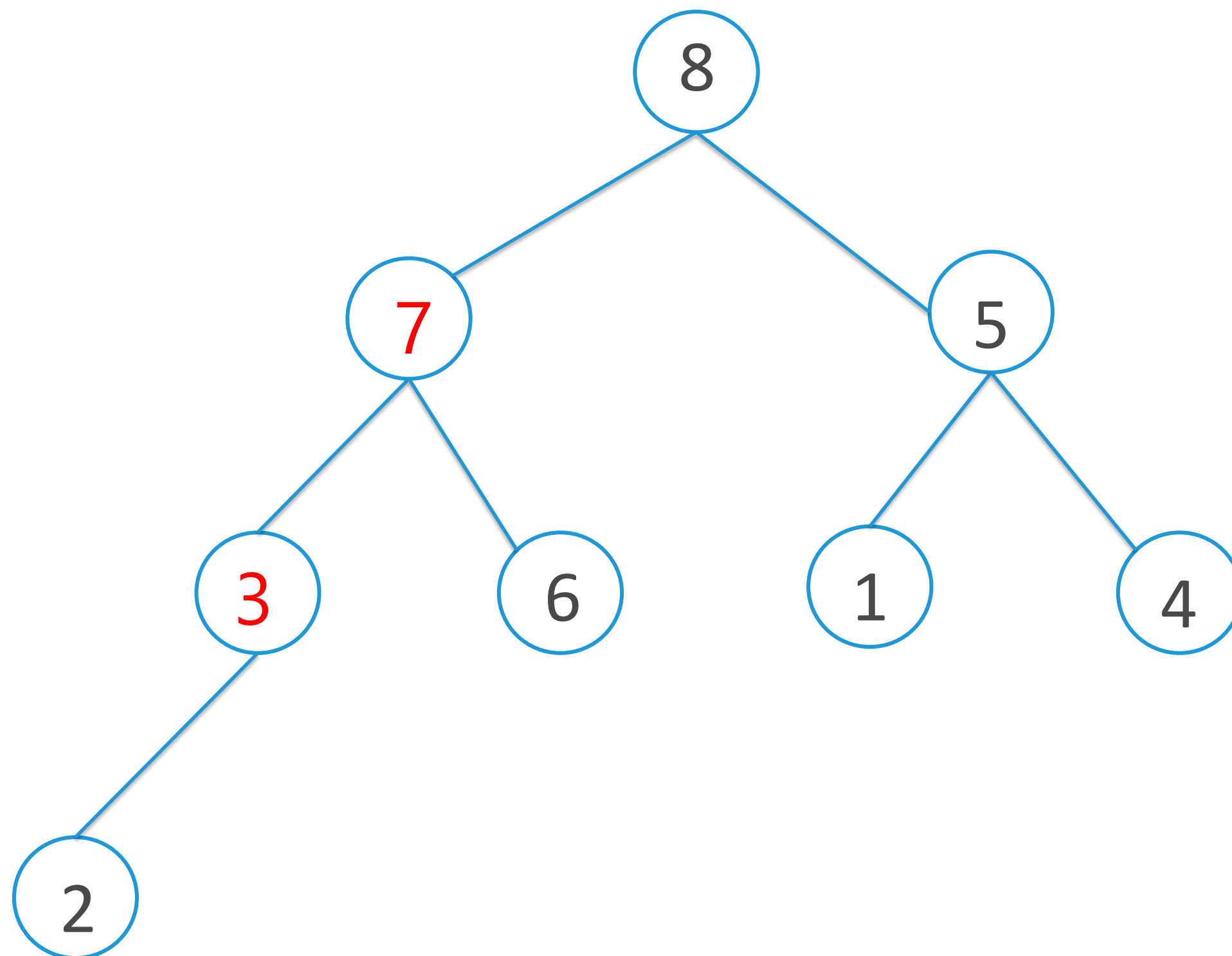
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 8 | 7 | 5 | 3 | 6 | 1 | 4 | 2 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Sharing is learning

I. Các thuật toán sắp xếp

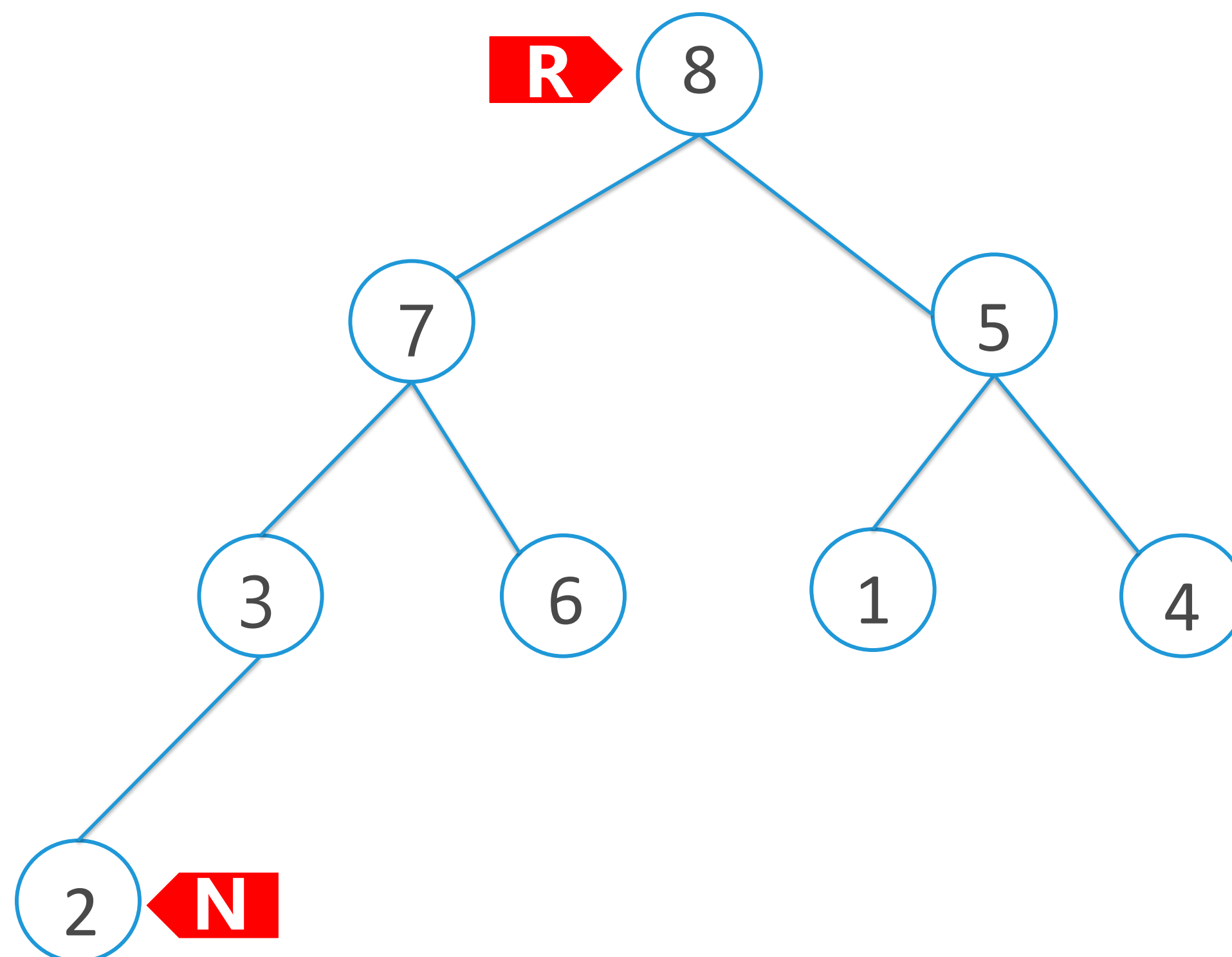


Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 8 | 7 | 5 | 3 | 6 | 1 | 4 | 2 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Hiệu chỉnh xong, tiến hành sắp xếp



Sharing is learning

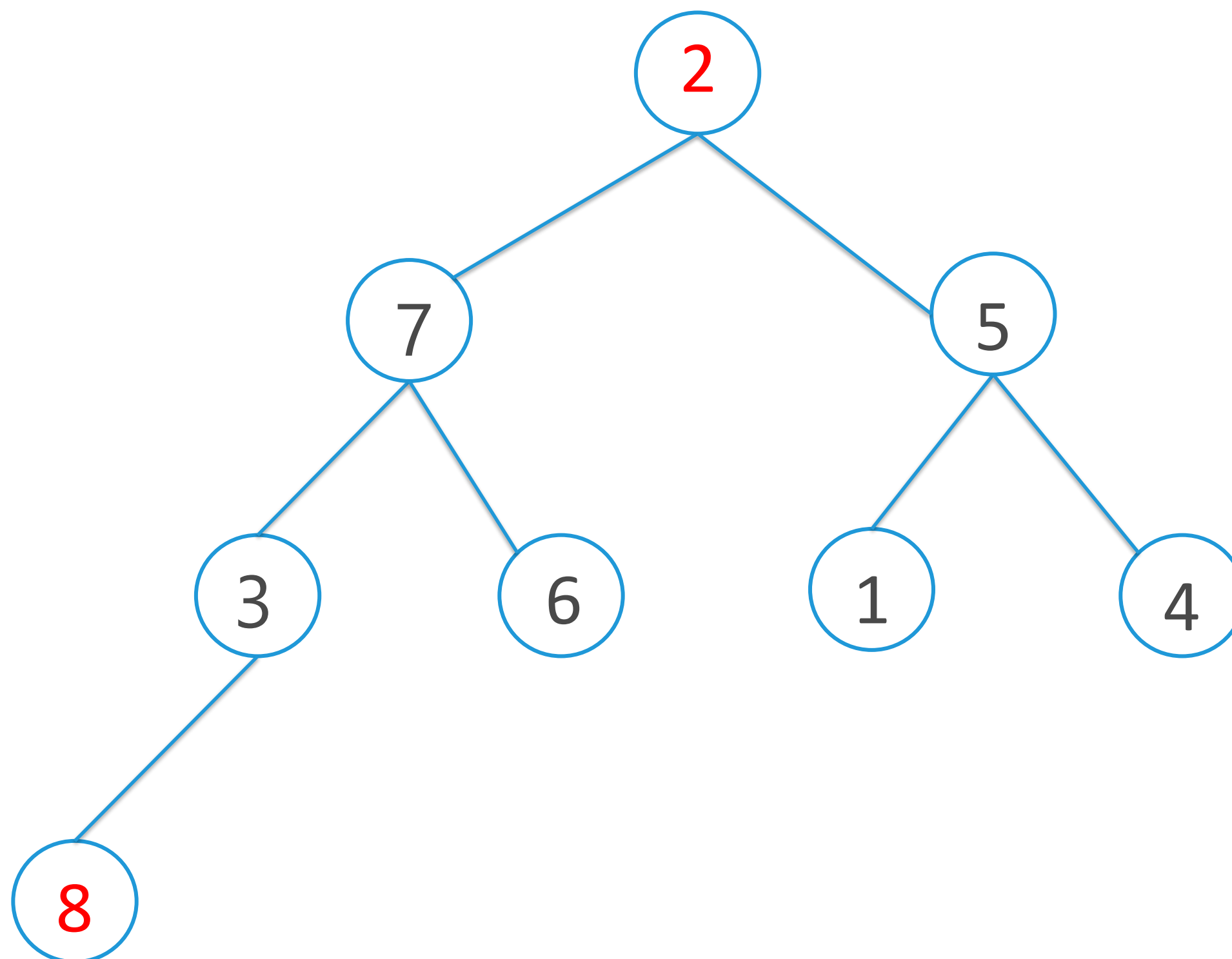
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 2 | 7 | 5 | 3 | 6 | 1 | 4 | 8 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Sharing is learning

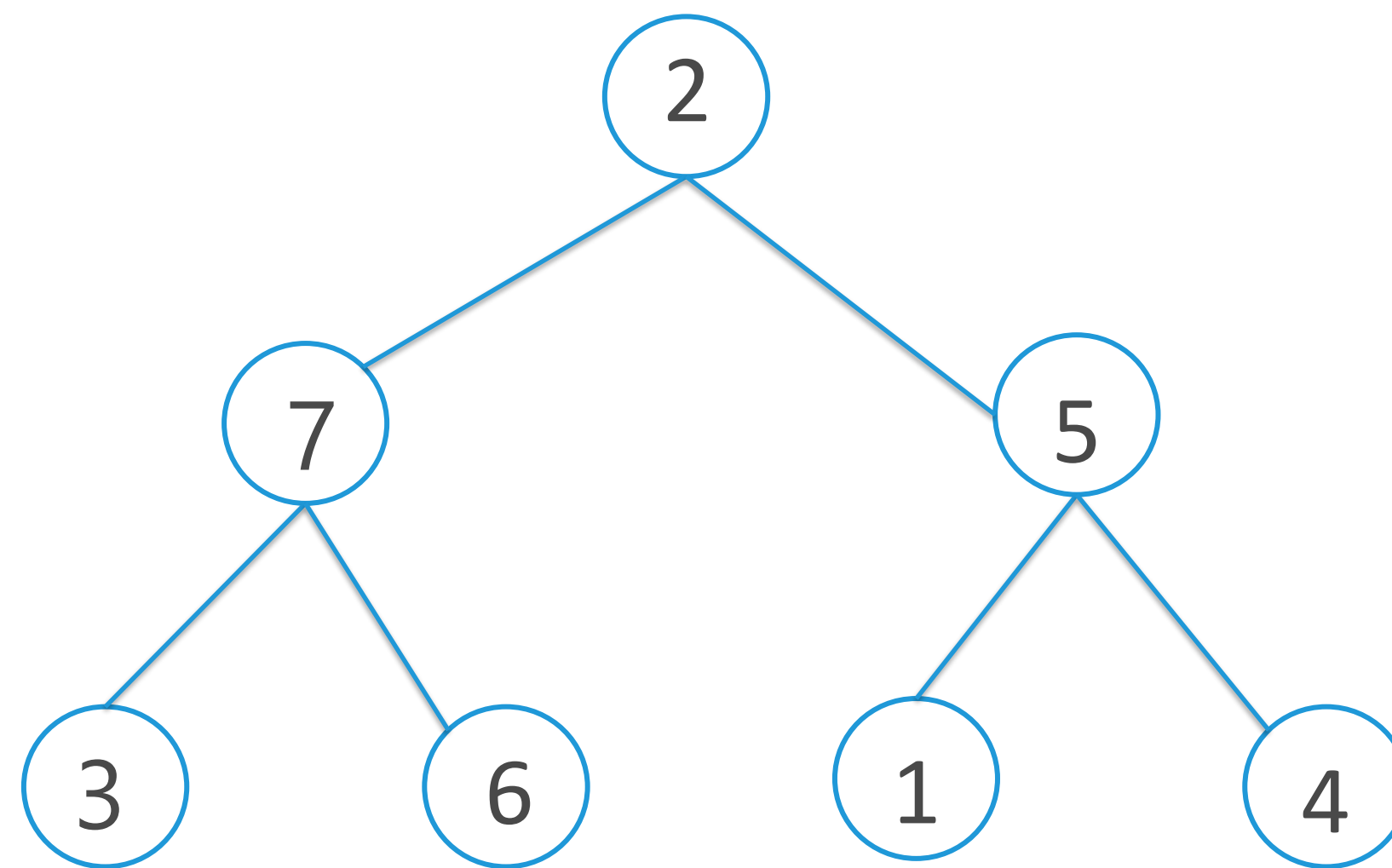
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 2 | 7 | 5 | 3 | 6 | 1 | 4 | 8 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Số phần tử còn lại: $7 > 1$, tiếp tục hiệu chỉnh dãy còn lại thành heap

Sharing is learning

I. Các thuật toán sắp xếp

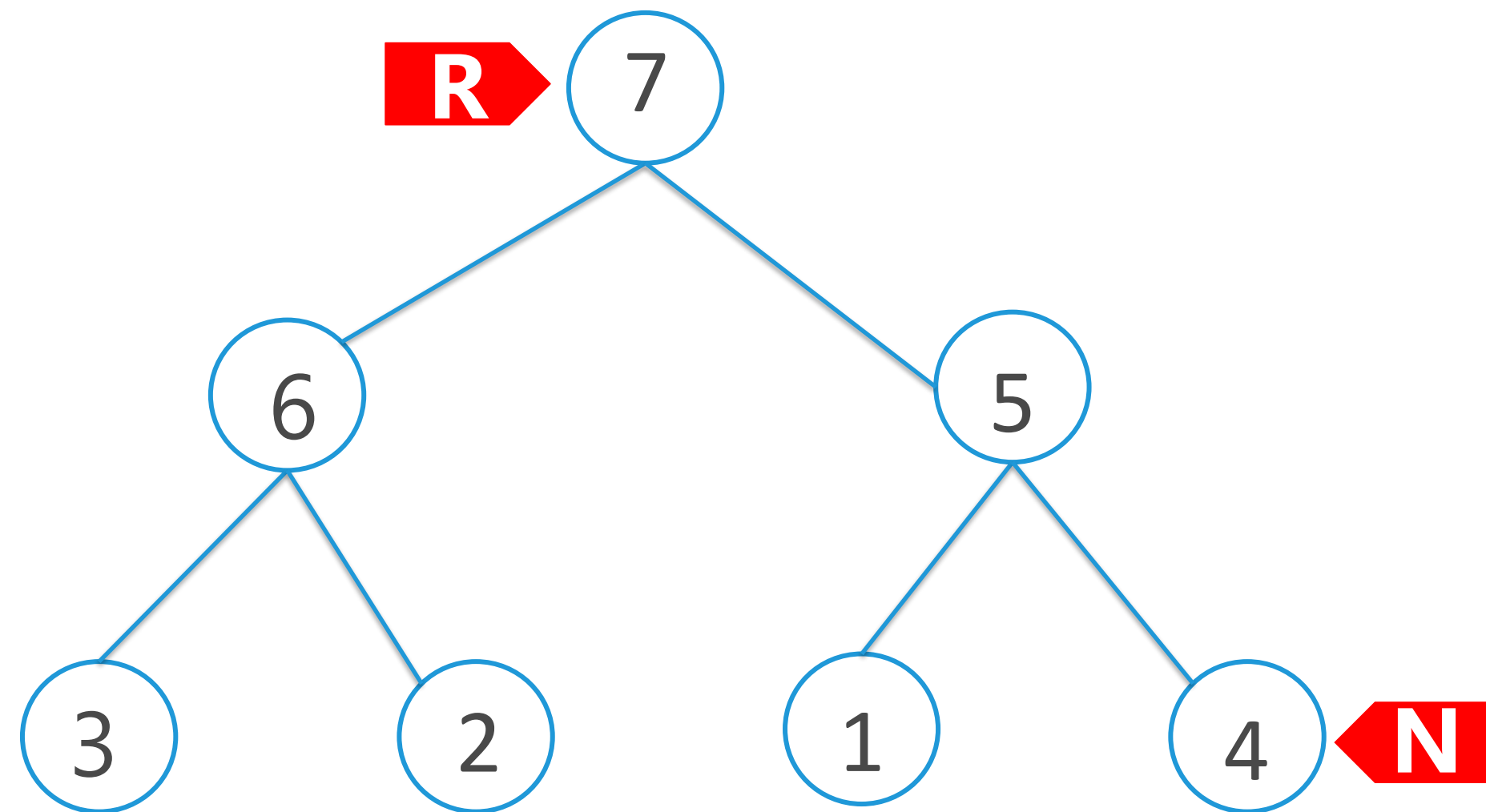


Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 7 | 6 | 5 | 3 | 2 | 1 | 4 | 8 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Hiệu chỉnh tương tự, ta thu được heap và tiến hành sắp xếp:



Sharing is learning

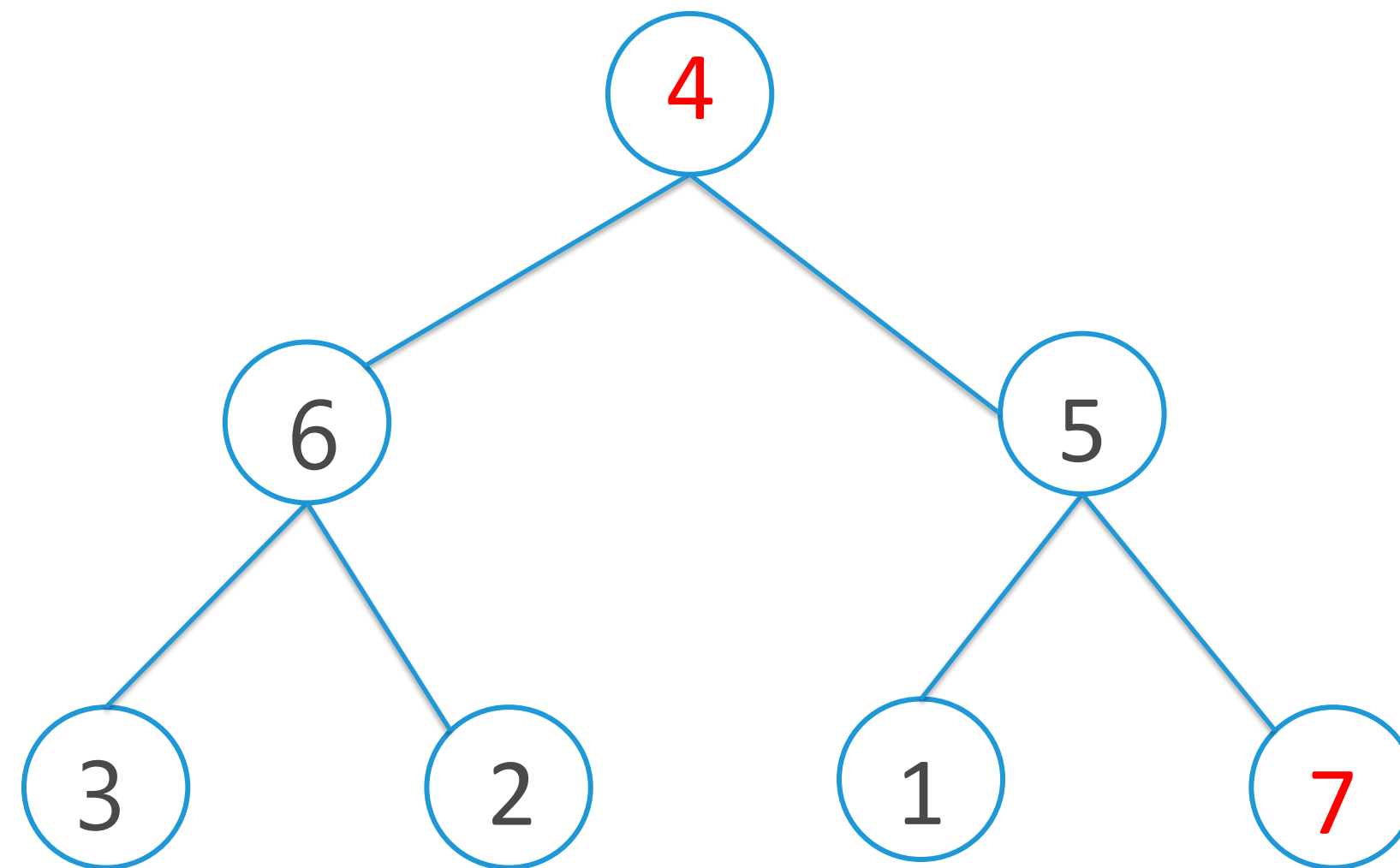
II. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 4 | 6 | 5 | 3 | 2 | 1 | 7 | 8 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Sharing is learning

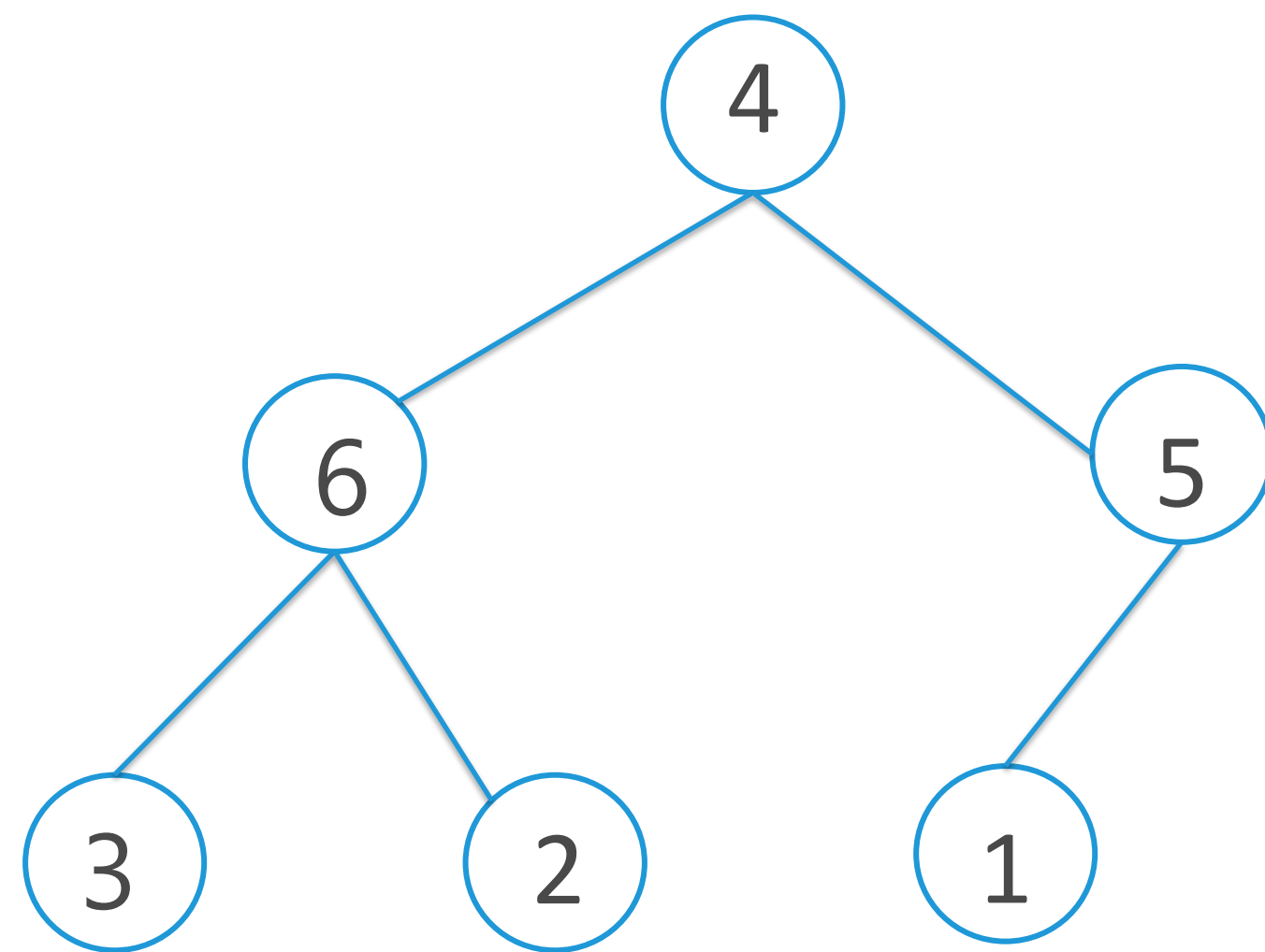
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 4 | 6 | 5 | 3 | 2 | 1 | 7 | 8 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Số phần tử còn lại: $6 > 1$, tiếp tục hiệu chỉnh dãy còn lại thành heap



Sharing is learning

I. Các thuật toán sắp xếp

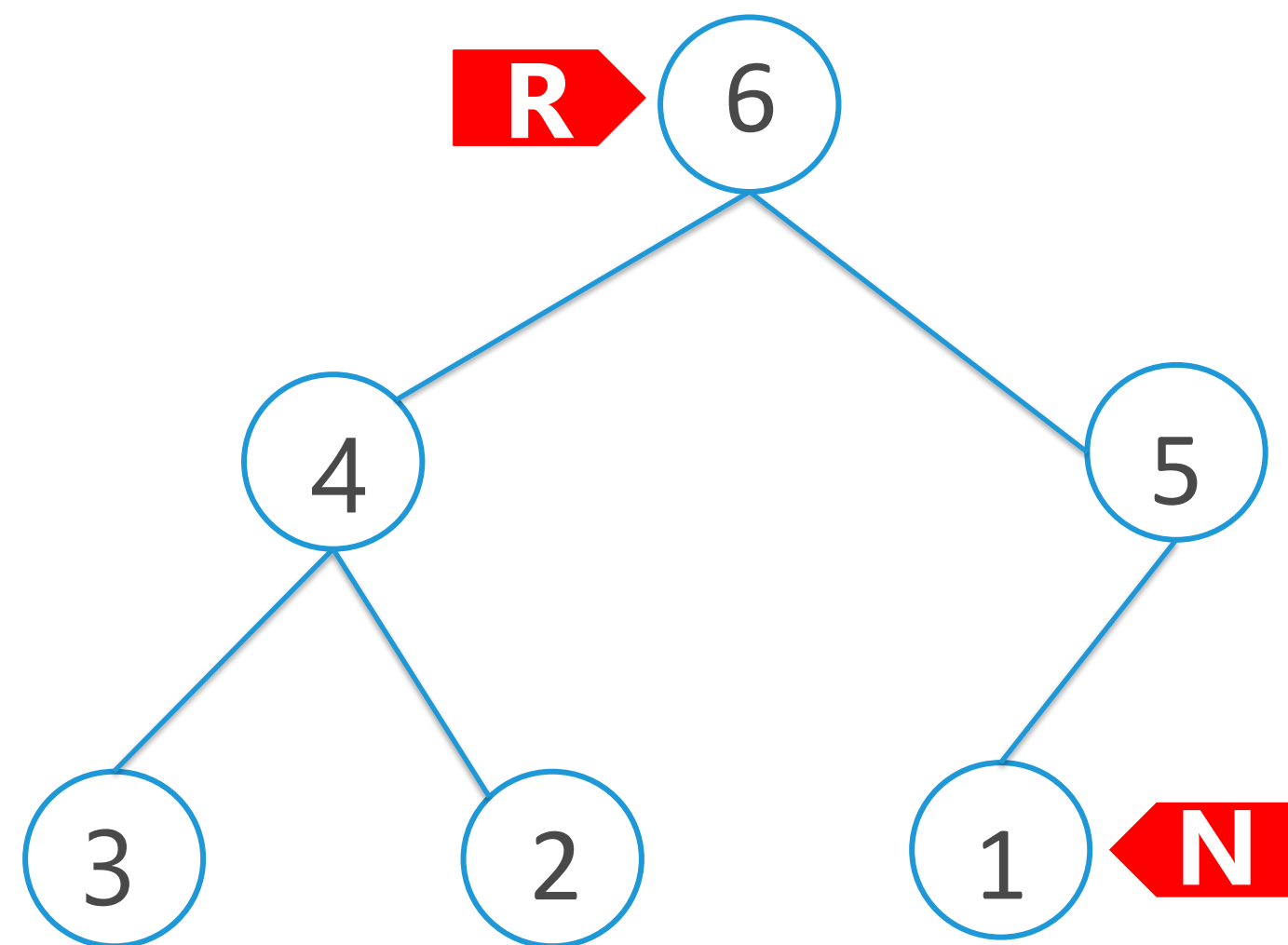


Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 6 | 4 | 5 | 3 | 2 | 1 | 7 | 8 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Hiệu chỉnh tương tự, ta thu được heap và tiến hành sắp xếp:



Sharing is learning

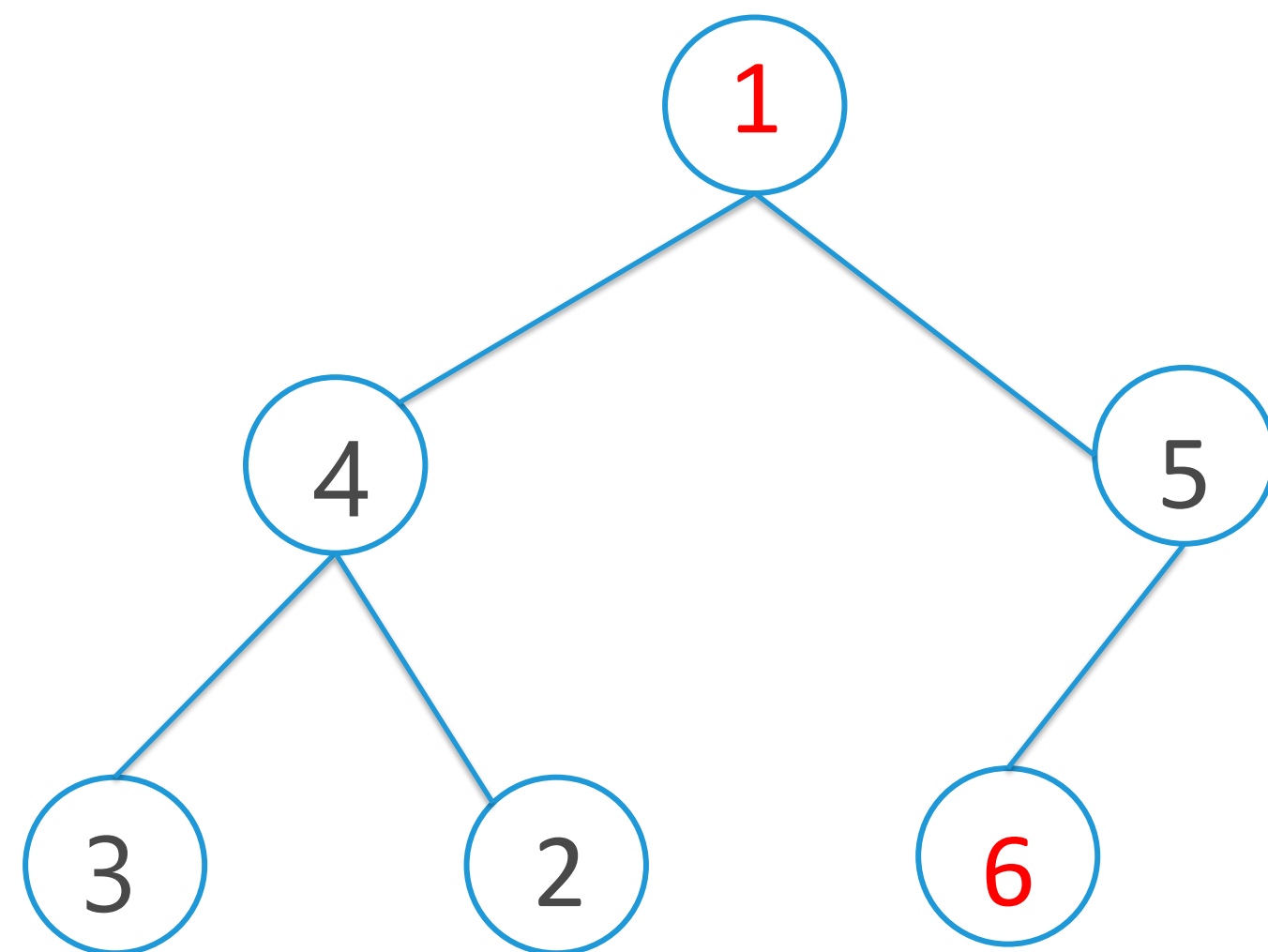
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 1 | 4 | 5 | 3 | 2 | 6 | 7 | 8 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Sharing is learning

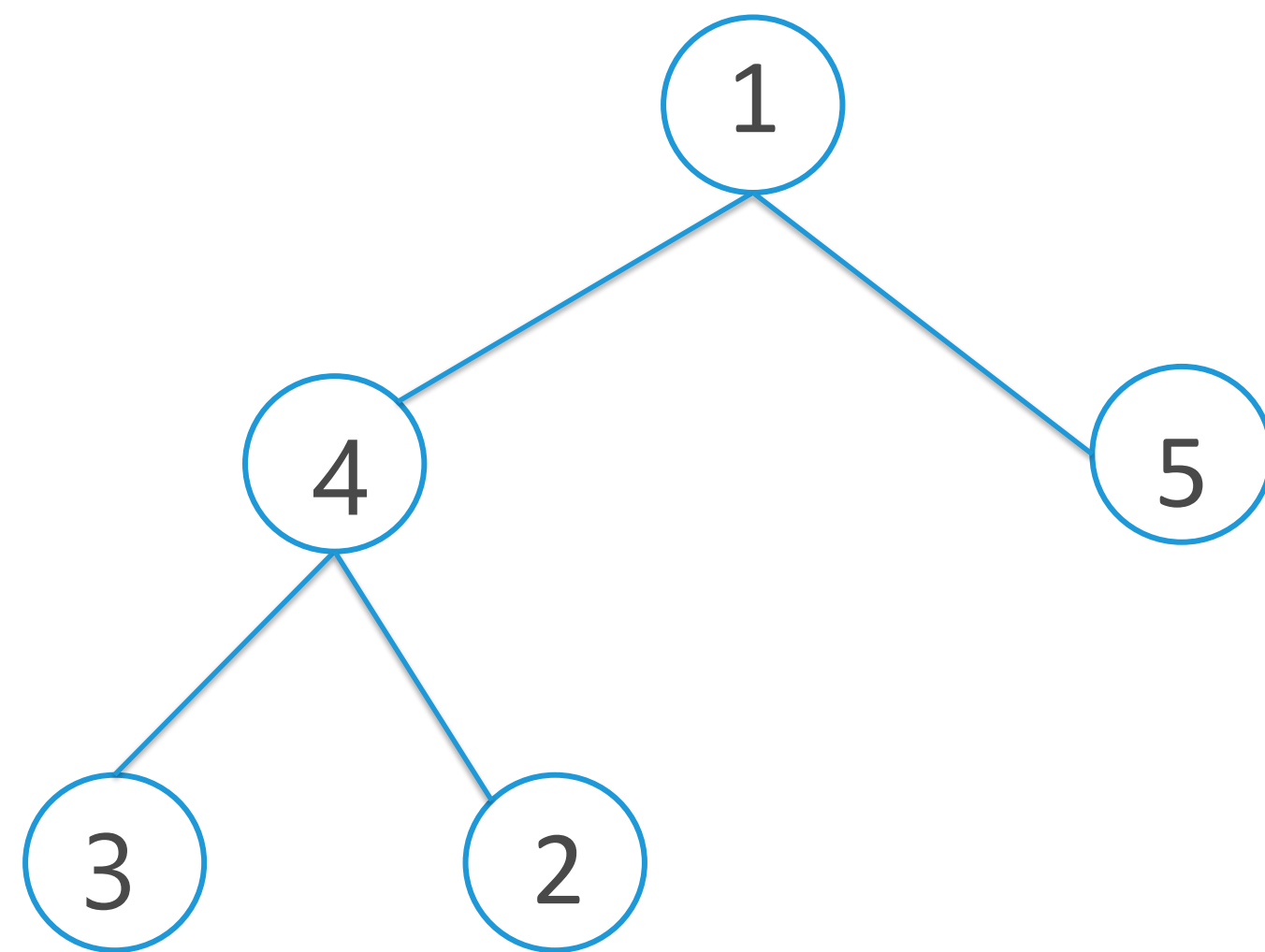
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 1 | 4 | 5 | 3 | 2 | 6 | 7 | 8 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Số phần tử còn lại: $5 > 1$, tiếp tục hiệu chỉnh dãy còn lại thành heap



Sharing is learning

I. Các thuật toán sắp xếp

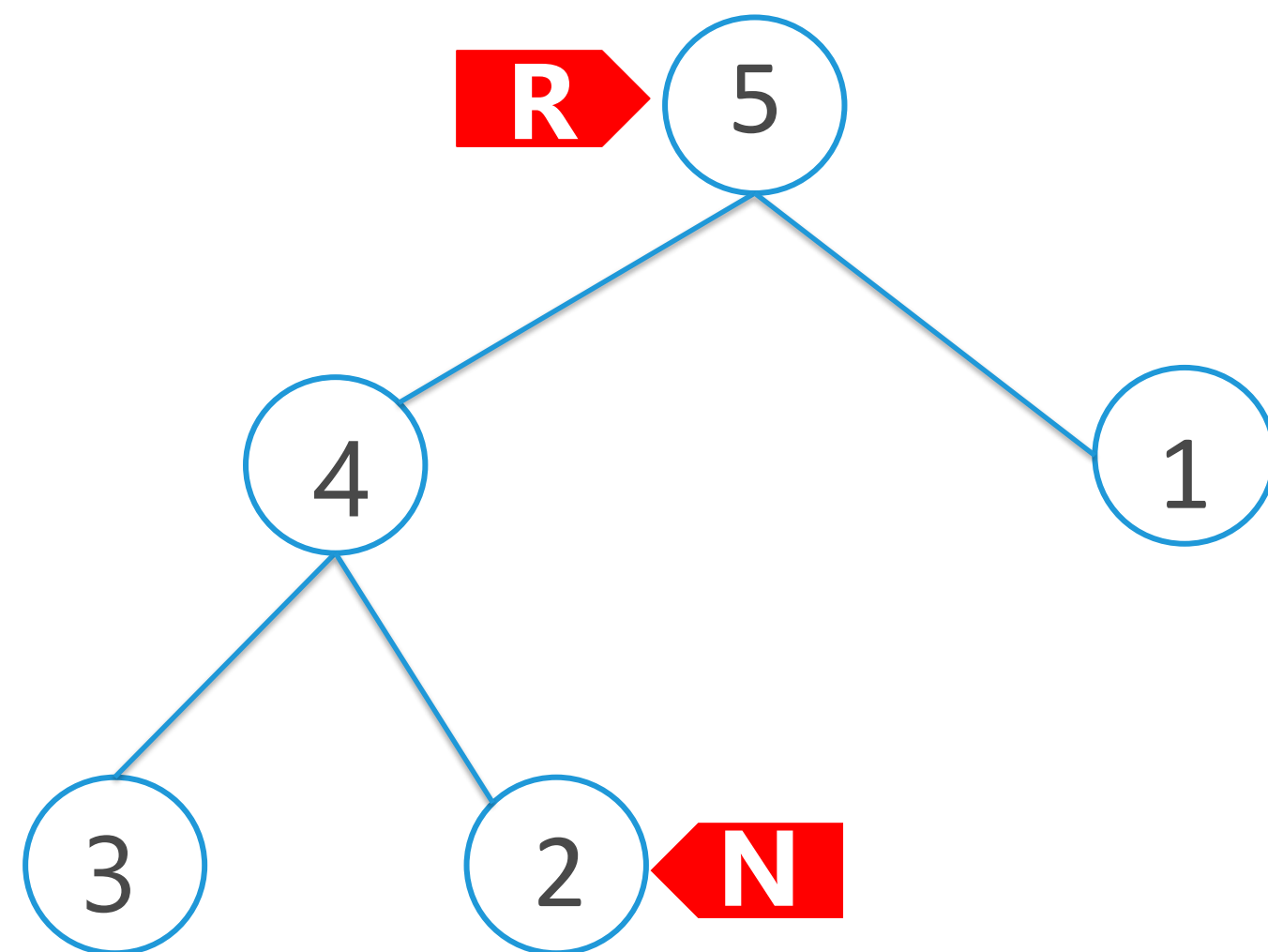


Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 5 | 4 | 1 | 3 | 2 | 6 | 7 | 8 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Hiệu chỉnh tương tự, ta thu được heap và tiến hành sắp xếp:



Sharing is learning

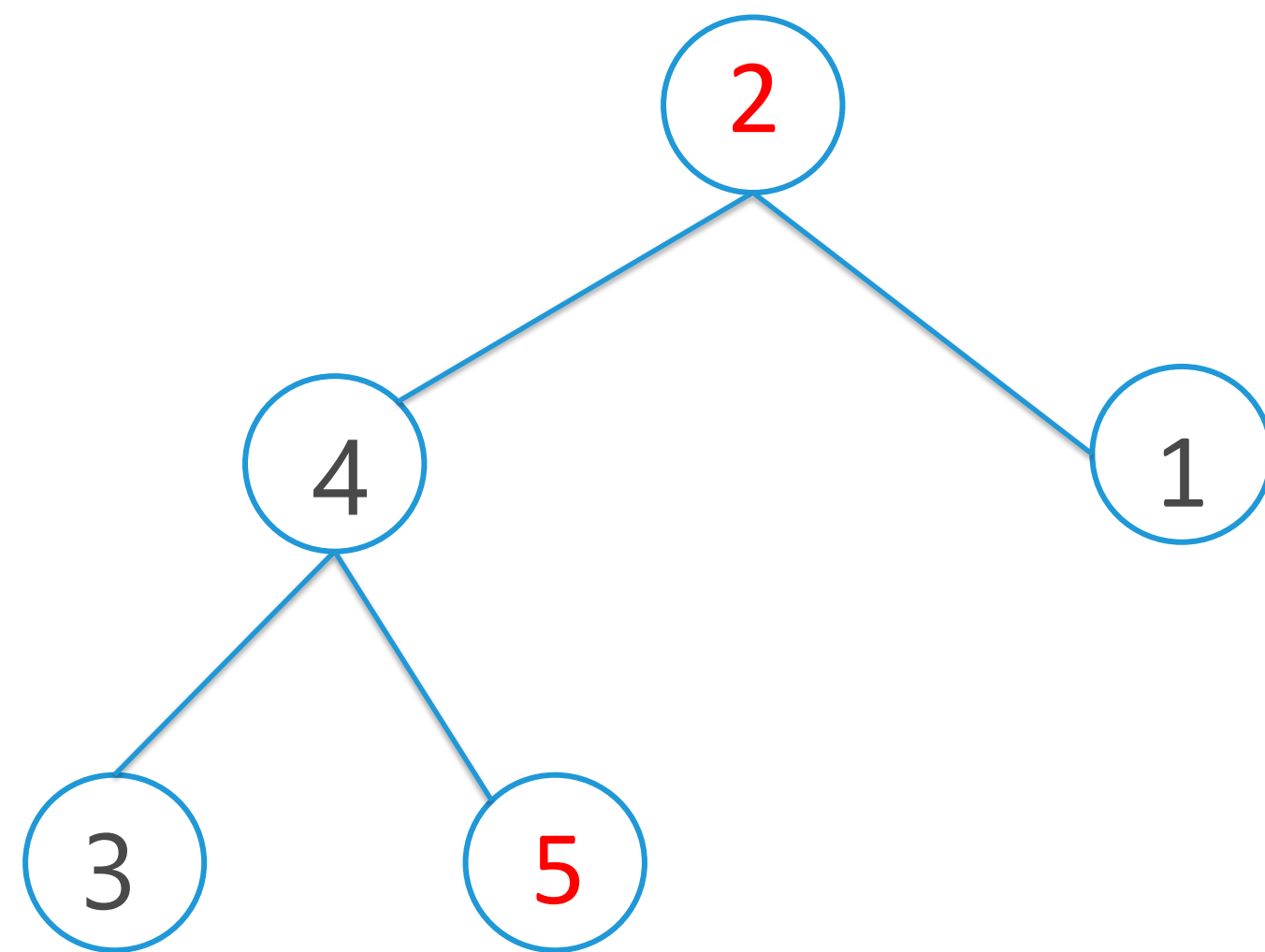
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 2 | 4 | 1 | 3 | 5 | 6 | 7 | 8 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Sharing is learning

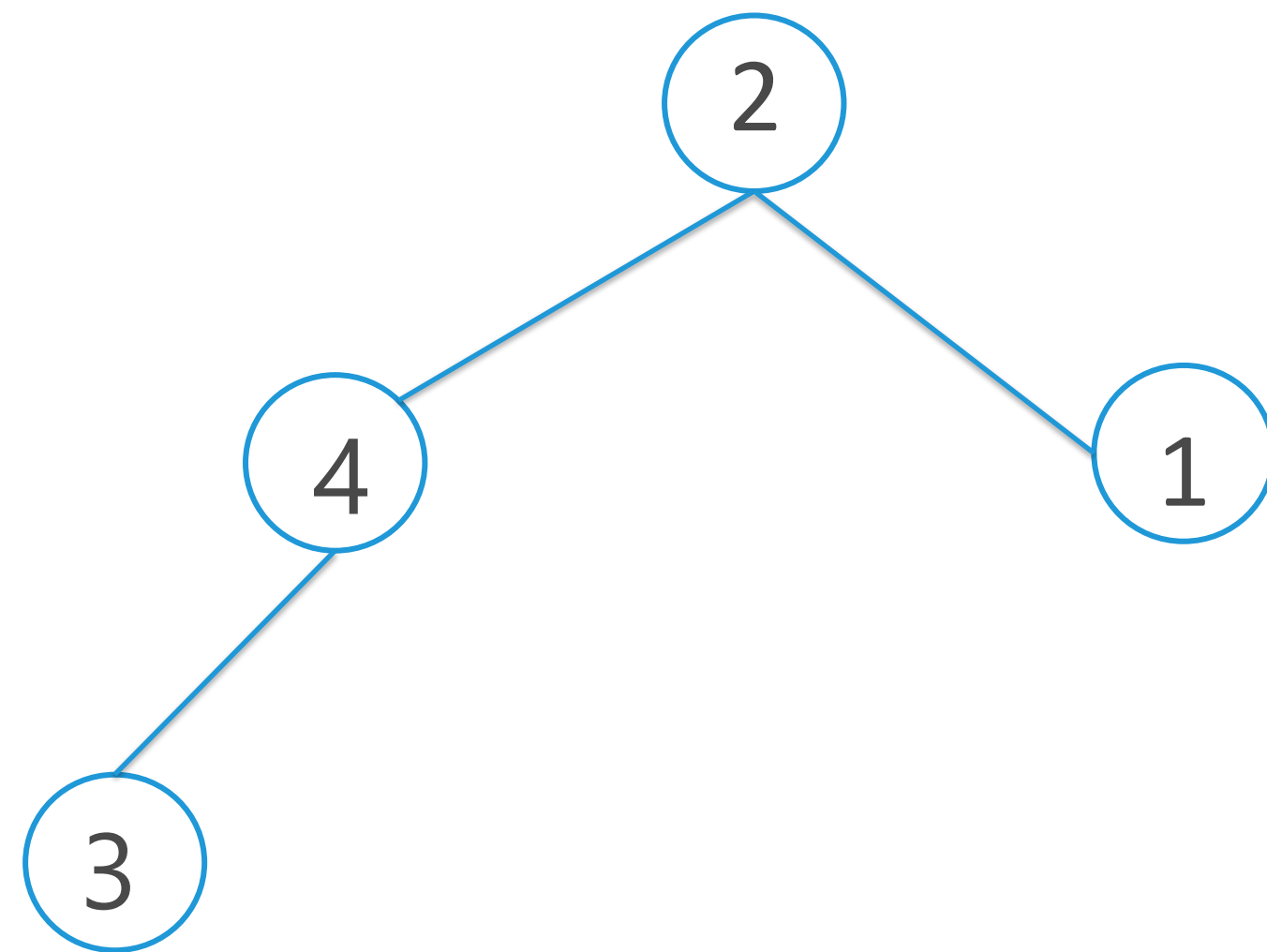
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 2 | 4 | 1 | 3 | 5 | 6 | 7 | 8 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Số phần tử còn lại: $4 > 1$, tiếp tục hiệu chỉnh dãy còn lại thành heap



Sharing is learning

I. Các thuật toán sắp xếp

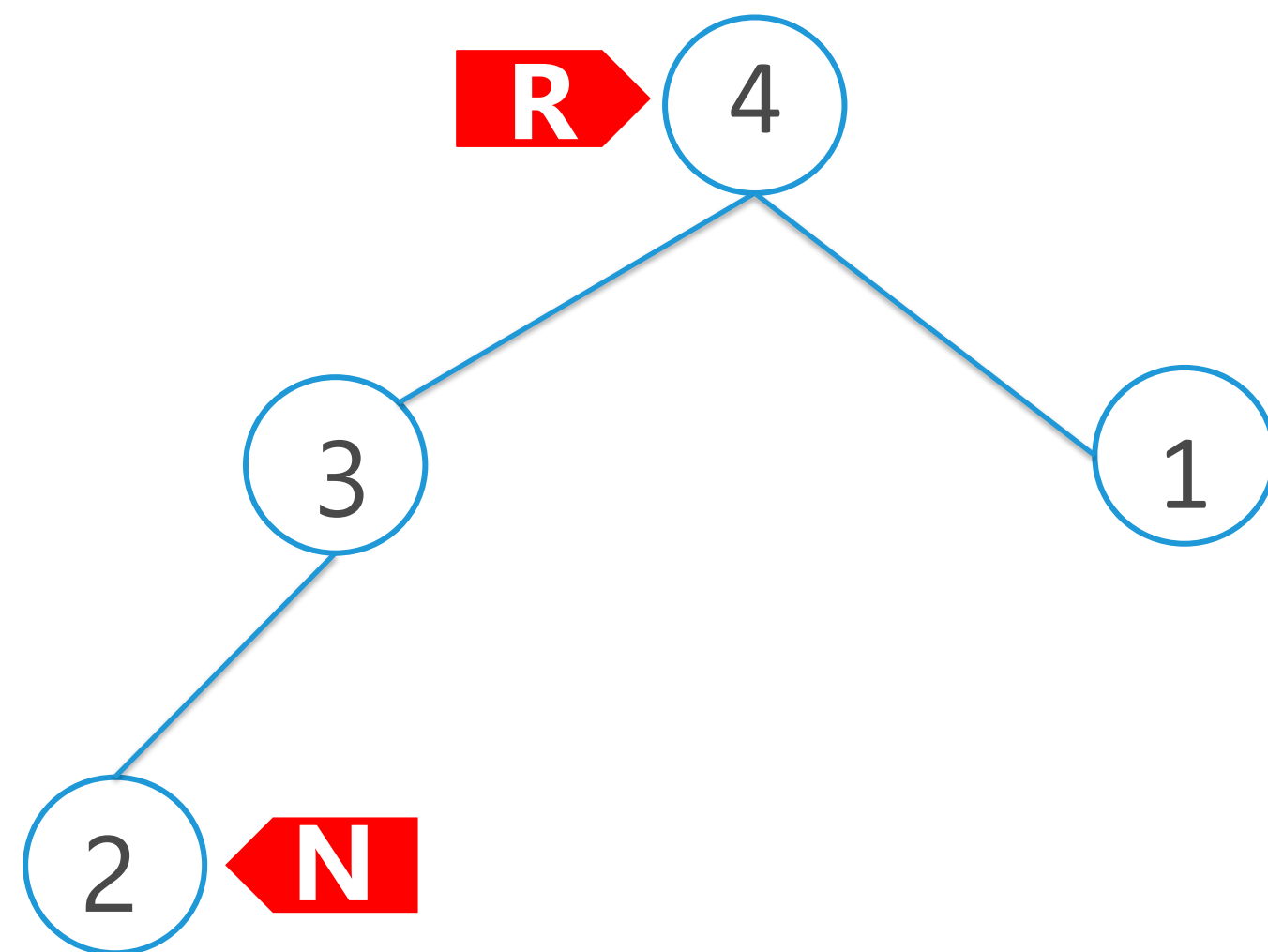


Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 4 | 3 | 1 | 2 | 5 | 6 | 7 | 8 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Hiệu chỉnh tương tự, ta thu được heap và tiến hành sắp xếp:



Sharing is learning

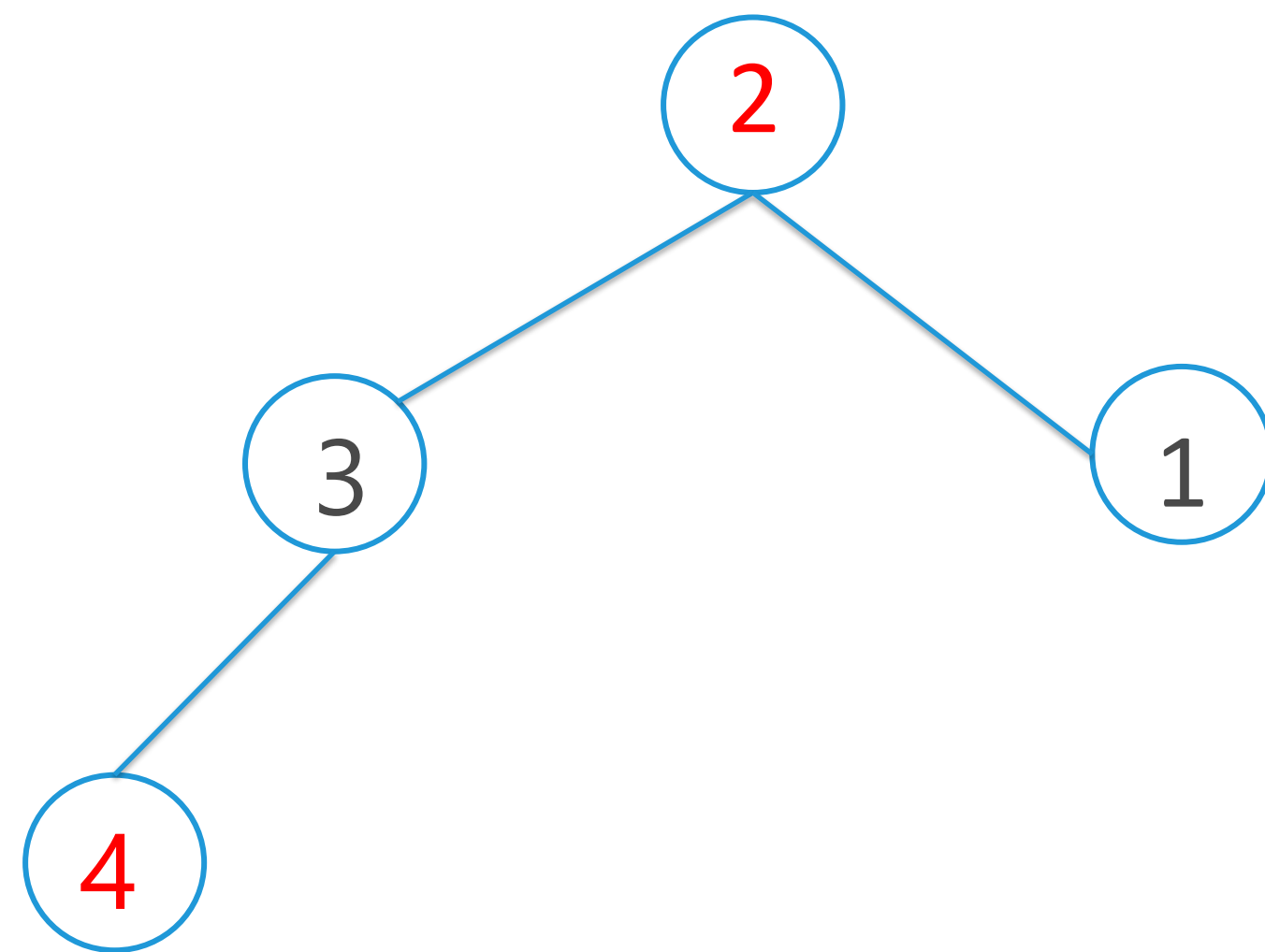
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 2 | 3 | 1 | 4 | 5 | 6 | 7 | 8 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Sharing is learning

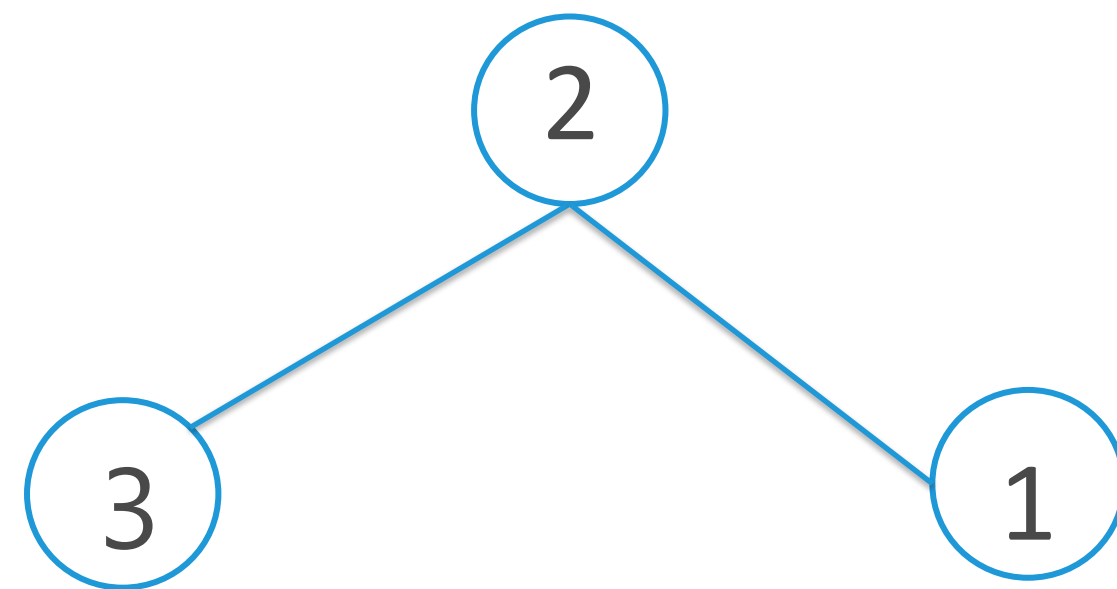
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 2 | 3 | 1 | 4 | 5 | 6 | 7 | 8 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Số phần tử còn lại: $3 > 1$, tiếp tục hiệu chỉnh dãy còn lại thành heap



Sharing is learning

I. Các thuật toán sắp xếp

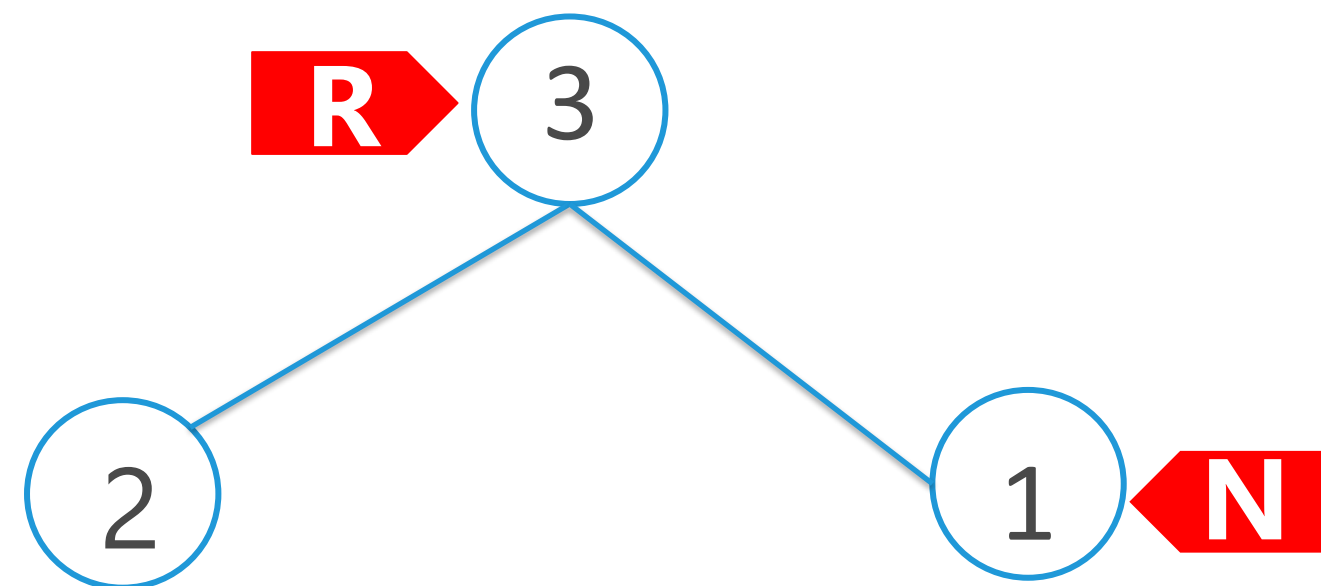


Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 3 | 2 | 1 | 4 | 5 | 6 | 7 | 8 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Hiệu chỉnh tương tự, ta thu được heap và tiến hành sắp xếp:

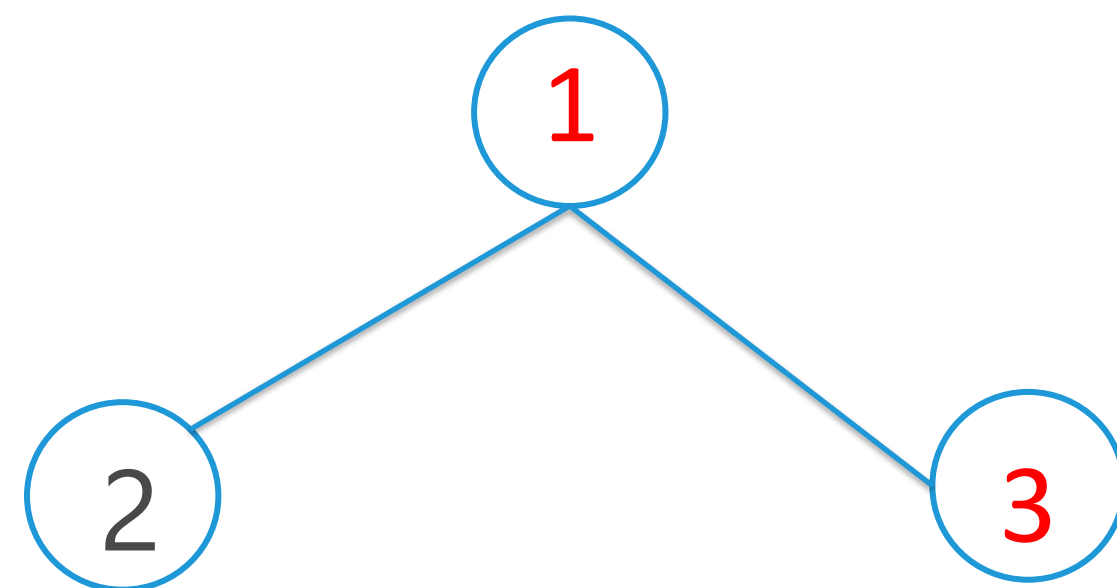


Sharing is learning

I. Các thuật toán sắp xếp

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Sharing is learning



Sharing is learning

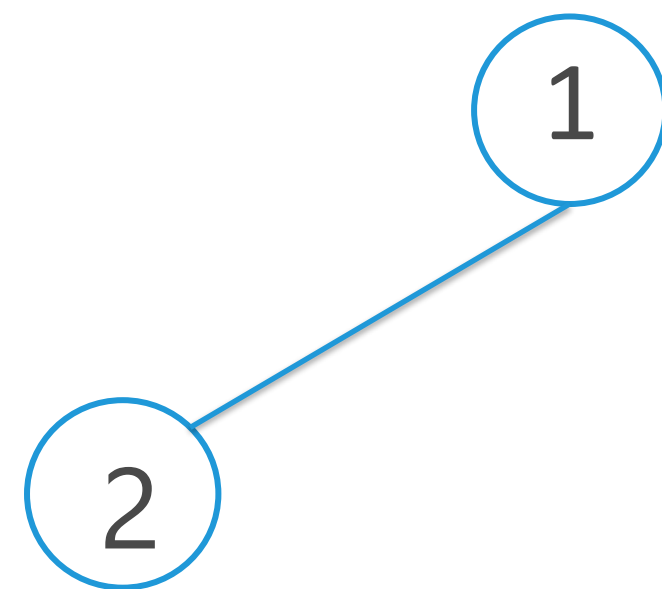
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Số phần tử còn lại: $3 > 1$, tiếp tục hiệu chỉnh dãy còn lại thành heap

Sharing is learning

I. Các thuật toán sắp xếp

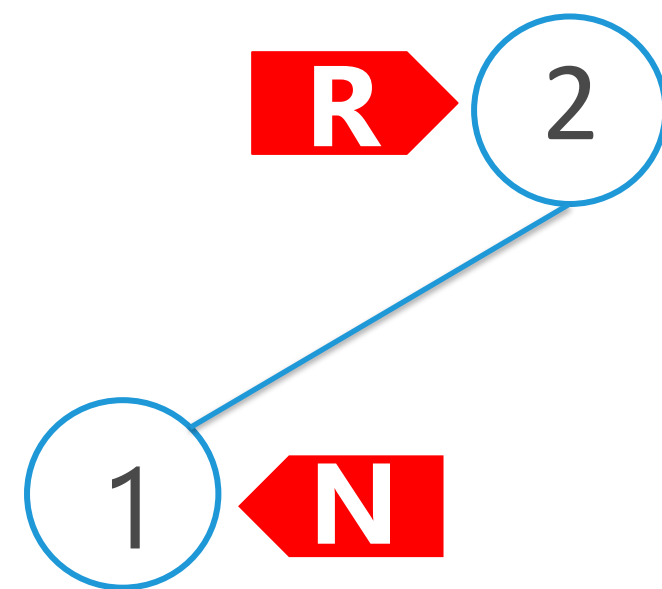


Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 2 | 1 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Hiệu chỉnh tương tự, ta thu được heap và tiến hành sắp xếp:



Sharing is learning

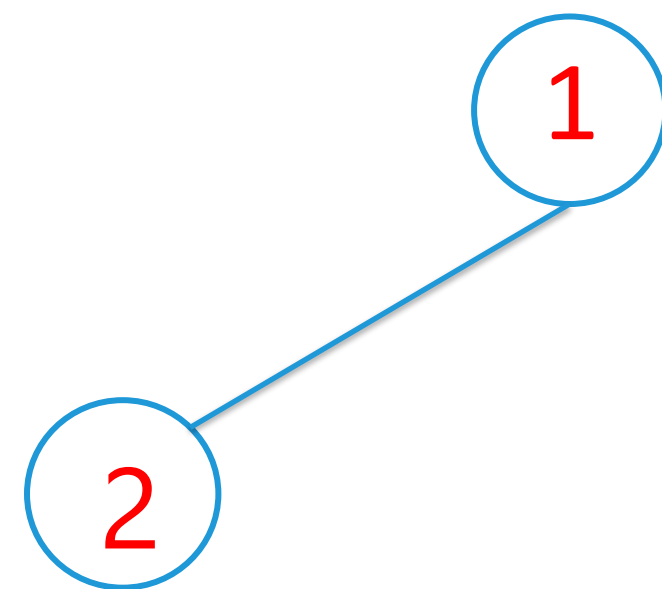
I. Các thuật toán sắp xếp



Sharing is learning

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



Sharing is learning

I. Các thuật toán sắp xếp

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

1

Số phần tử còn lại: $1=1$, dừng



Sharing is learning



Sharing is learning

I. Các thuật toán sắp xếp

5. Heap sort

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| A: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| VT: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Ta có dãy thu được sau khi đã sắp xếp.



Sharing is learning



Sharing is learning

I. Các thuật toán sắp xếp

5. Heap sort

Ví dụ tham khảo:

6 5 3 1 8 7 2 4



Sharing is learning



Sharing is learning

I. Các thuật toán sắp xếp

5. Heap sort

Độ phức tạp :

- Best case: **$O(n \log n)$**
- Worst case: **$O(n \log n)$**
- Average case: **$O(n \log n)$**



Sharing is learning



Sharing is learning

I. Các thuật toán sắp xếp

Một số bài tập tham khảo:

Câu 1: Độ phức tạp trung bình của thuật toán sắp xếp nhanh (Quick Sort) là :

- a. $O(n \log n)$
- b. $O(n^2)$
- c. $O(\log n)$
- d. $O(n)$



Sharing is learning



Sharing is learning

I. Các thuật toán sắp xếp

Một số bài tập tham khảo:

Câu 2: Thuật toán nào sau đây dựa trên giải thuật Chia để trị?

- a. Insertion sort
- b. Merge sort
- c. Heap sort
- d. Selection sort



Sharing is learning



Sharing is learning

I. Các thuật toán sắp xếp

Một số bài tập tham khảo:

Câu 3: Ý nào sau đây là nhược điểm của thuật toán sắp xếp Insertion sort?

- a. Cần dùng thêm bộ nhớ để lưu mảng
- b. Không đủ nhanh với dữ liệu lớn
- c. Cả 2 ý trên đều đúng
- d. Cả 2 ý trên đều sai



Sharing is learning



Sharing is learning

I. Các thuật toán sắp xếp

Một số bài tập tham khảo:

Câu 4: Nhận xét nào sau đây đúng về thuật toán sắp xếp Merge sort?

- a. Không tối ưu bộ nhớ
- b. Nhanh hơn Quick sort
- c. Không nhận dạng được mảng đã sắp xếp
- d. Chỉ có a, c đúng
- e. Cả a, b, c đều đúng



Sharing is learning



Sharing is learning

I. Các thuật toán sắp xếp

Một số bài tập tham khảo:

Câu 5: Giả sử cần sắp xếp dãy

$A = \{50, 17, 15, 9, 62, 86, 12, 14, 32, 6\}$ giảm dần thì dãy thứ 2 thu được sau khi tiến hành chạy thuật toán sắp xếp Insertion sort là gì?

86 62 50 17 15 9 12 14 32 6



Sharing is learning



Sharing is learning

I. Các thuật toán sắp xếp



Sharing is learning

Tham khảo: bảng thống kê một số thông số của các thuật toán sắp xếp (được tổng hợp từ Wikipedia)

| STT | Thuật toán | Độ phức tạp | | | Bộ nhớ | Stable |
|-----|-----------------------|---------------|-------------------------|---------------|-------------|----------|
| | | Tốt nhất | Trung bình | Xấu nhất | | |
| 1 | Bubble Sort | $O(n)$ | $O(n^2)$ | $O(n^2)$ | $O(1)$ | Có |
| 2 | Shaker Sort | $O(n)$ | $O(n^2)$ | $O(n^2)$ | $O(1)$ | Có |
| 3 | Selection Sort | $O(n^2)$ | $O(n^2)$ | $O(n^2)$ | $O(1)$ | Không |
| 4 | Insertion Sort | $O(n)$ | $O(n^2)$ | $O(n^2)$ | $O(1)$ | Có |
| 5 | Binary Insertion Sort | $O(n)$ | $O(n^2)$ | $O(n^2)$ | $O(1)$ | Có |
| 6 | Shell Sort | $O(n \log n)$ | depends on gap sequence | $O(n^2)$ | $O(1)$ | Không |
| 7 | Heap Sort | $O(n \log n)$ | $O(n \log n)$ | $O(n \log n)$ | $O(1)$ | Không |
| 8 | Merge Sort | $O(n \log n)$ | $O(n \log n)$ | $O(n \log n)$ | $O(n)$ | Có |
| 9 | Quick Sort | $O(n \log n)$ | $O(n \log n)$ | $O(n^2)$ | $O(\log n)$ | Có/Không |
| 10 | Counting Sort | $O(n+k)$ | $O(n+k)$ | $O(n+k)$ | $O(n+k)$ | Có |
| 11 | Radix Sort | $O(kn)$ | $O(nk)$ | $O(nk)$ | $O(n+k)$ | Có |
| 12 | Flash Sort | $O(n)$ | $O(n+r)$ | $O(n^2)$ | $O(m)$ | Không |

Sharing is learning

I. Các thuật toán sắp xếp



Tham khảo thêm: chuỗi bài viết Study with me môn CTDL> BHT CNPM

- ✓ Phần 1: <https://tinyurl.com/4cmw6akt>
- ✓ Phần 2: <https://tinyurl.com/tynufdf9>
- ✓ Phần 3: <https://tinyurl.com/kxvb47m9>



Sharing is learning

I. Các thuật toán sắp xếp



Sharing is learning

Tổng hợp code tham khảo 5 các thuật toán sắp xếp trên:

<https://tinyurl.com/85ahprwh>

<https://tinyurl.com/3rah293v>



Sharing is learning

Nội dung Training

I. Các thuật toán sắp xếp

II. Cây nhị phân



Sharing is learning



Sharing is learning

II. Cây nhị phân

1. Khái niệm cây – cây nhị phân
2. Cây nhị phân tìm kiếm
3. Đặc điểm cây nhị phân tìm kiếm
4. Các thao tác trên cây nhị phân tìm kiếm

II. Cây nhị phân



1. Khái niệm cây – cây nhị phân

- Tóm tắt: Trong CTDL, **cây** là một cấu trúc dữ liệu **gồm các nút** (Node) được **liên kết** với nhau **theo quan hệ cha – con**. **Cây bắt đầu từ nút gốc** (Root node).
- Nút:
 - Nút **gốc** (Root node): Nút **không có** nút **cha**. (Đây là nút đặc biệt, khởi đầu của một cây)
 - Nút **lá**: Nút **không có** nút **con**.
 - Nút trong (nút nhánh): Nút có con.

II. Cây nhị phân



1. Khái niệm cây – cây nhị phân

- Bậc của nút: Số cây con của nút.
- Bậc của cây: Bậc lớn nhất của các nút trong cây.
- Mức của nút: Mức của cha của nút + 1
 - Mức (nút gốc) = 0.
 - Mức (nút T_i) = Mức (nút cha của T_i) + 1.
- Chiều cao cây: Mức lớn nhất của các nút trong cây.
- Độ dài đường đi từ gốc đến nút: Số nhánh cần đi qua kể từ gốc đến nút.

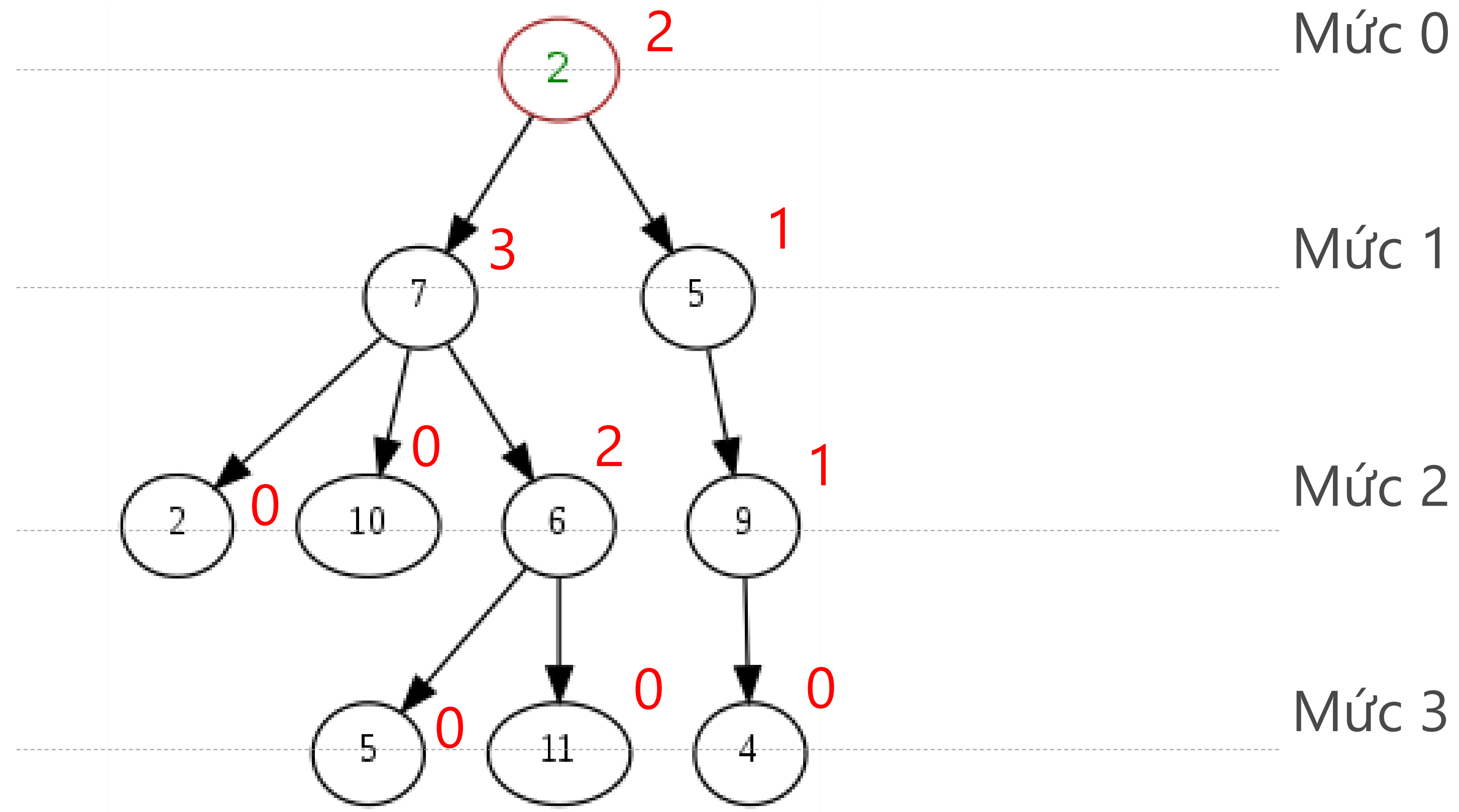
II. Cây nhị phân



Sharing is learning

1. Khái niệm cây – cây nhị phân

Ví dụ tổ chức dạng cây:



II. Cây nhị phân

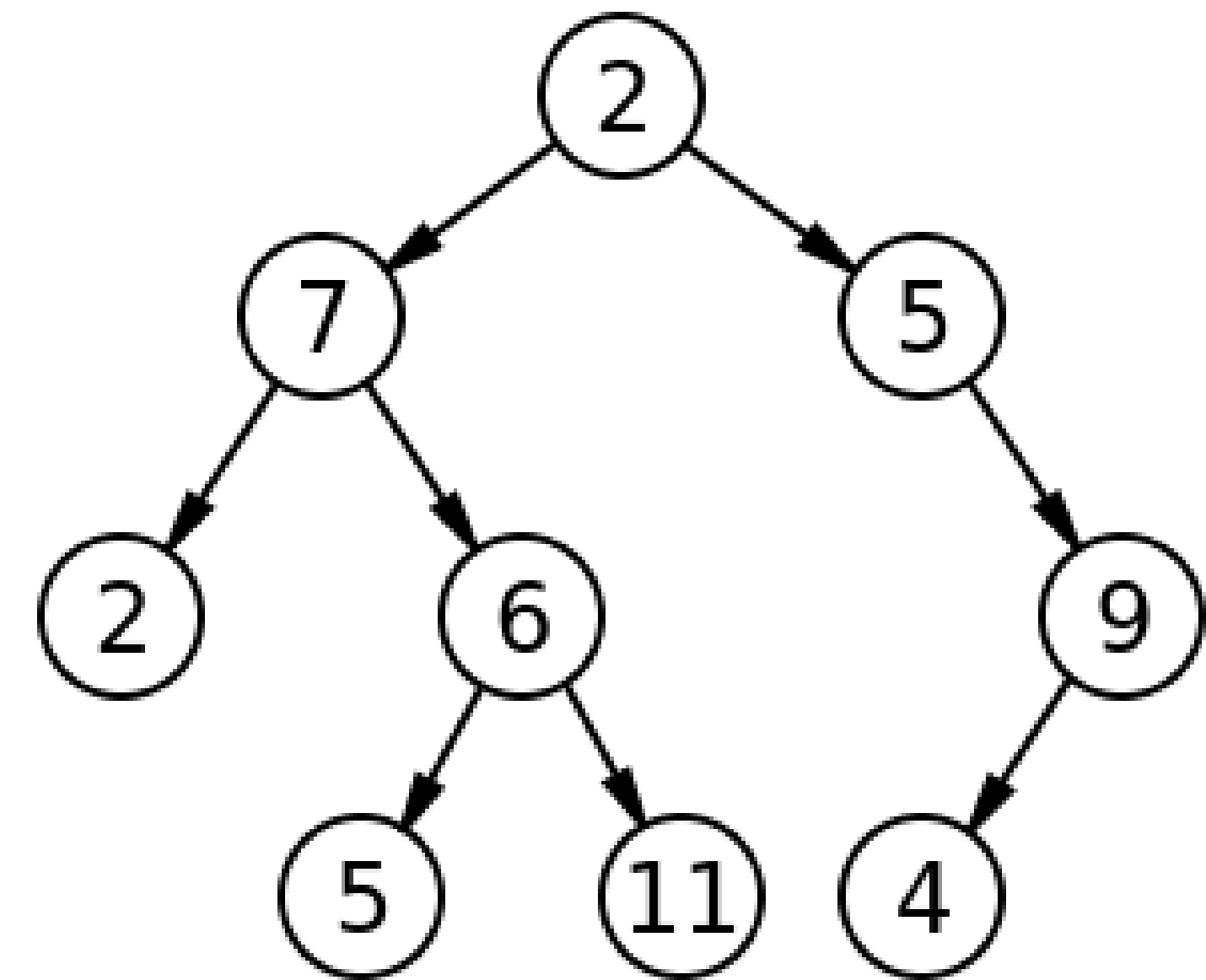


Sharing is learning

1. Khái niệm cây – cây nhị phân

Cây nhị phân:

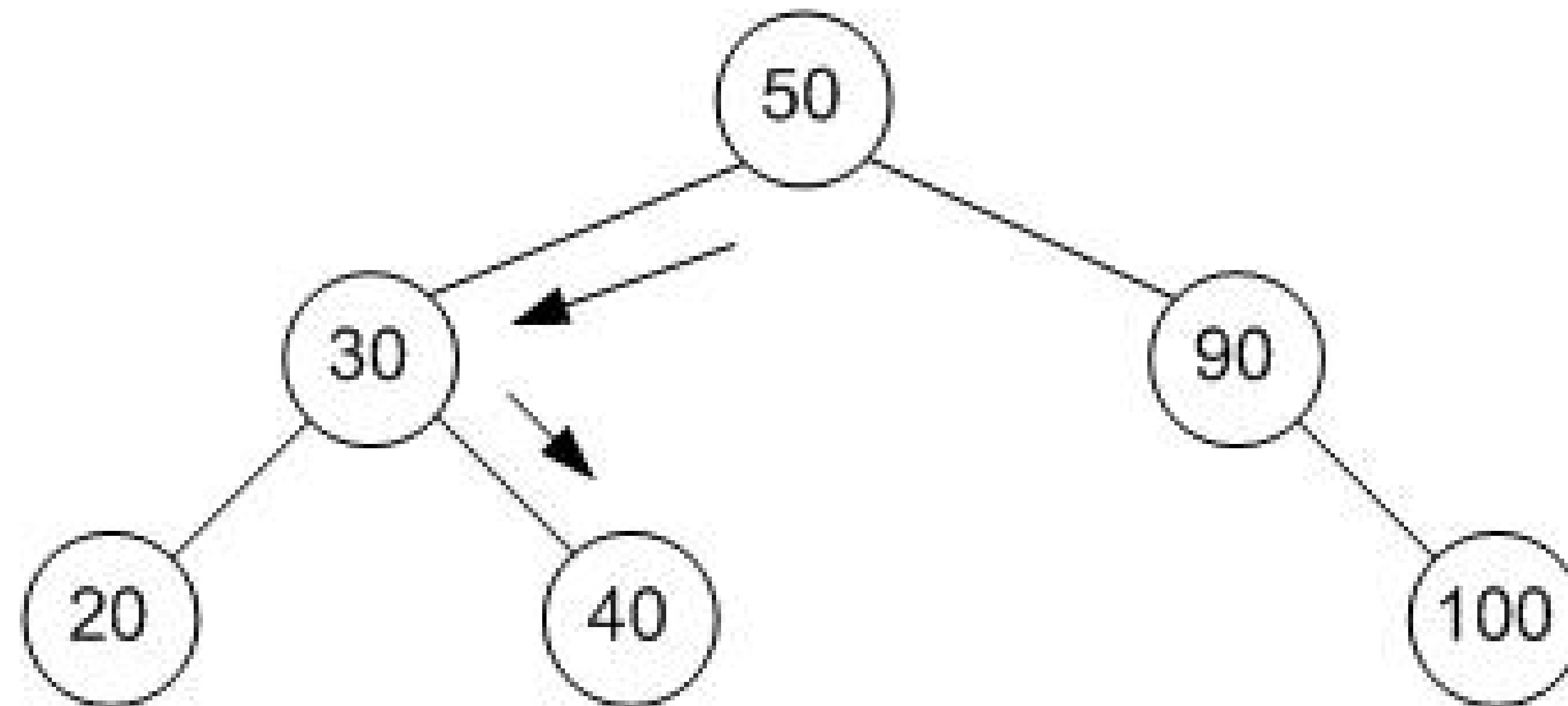
- Các tính chất của cây, cây nhị phân đều có.
- Số nút con mà mỗi nút có tối đa là 2.



II. Cây nhị phân

2. Cây nhị phân tìm kiếm (Binary search tree - BST)

- Các tính chất của cây nhị phân, cây nhị phân tìm kiếm đều có.
- Tuân theo **qui tắc** sau: Với **mỗi nút x** , **các nút ở cây con trái nhỏ hơn nút x** , **các nút ở cây con phải lớn hơn nút x** .



II. Cây nhị phân



Sharing is learning

3. Đặc điểm cây nhị phân tìm kiếm:

- Có thứ tự.
- Dễ tạo, thao tác sắp xếp và tìm kiếm dễ dàng.
- Không chứa phần tử trùng.
- Nút có **giá trị lớn nhất** nằm ở **nút phải nhất của cây**.
- Nút có **giá trị nhỏ nhất** nằm ở **nút trái nhất của cây**.

II. Cây nhị phân



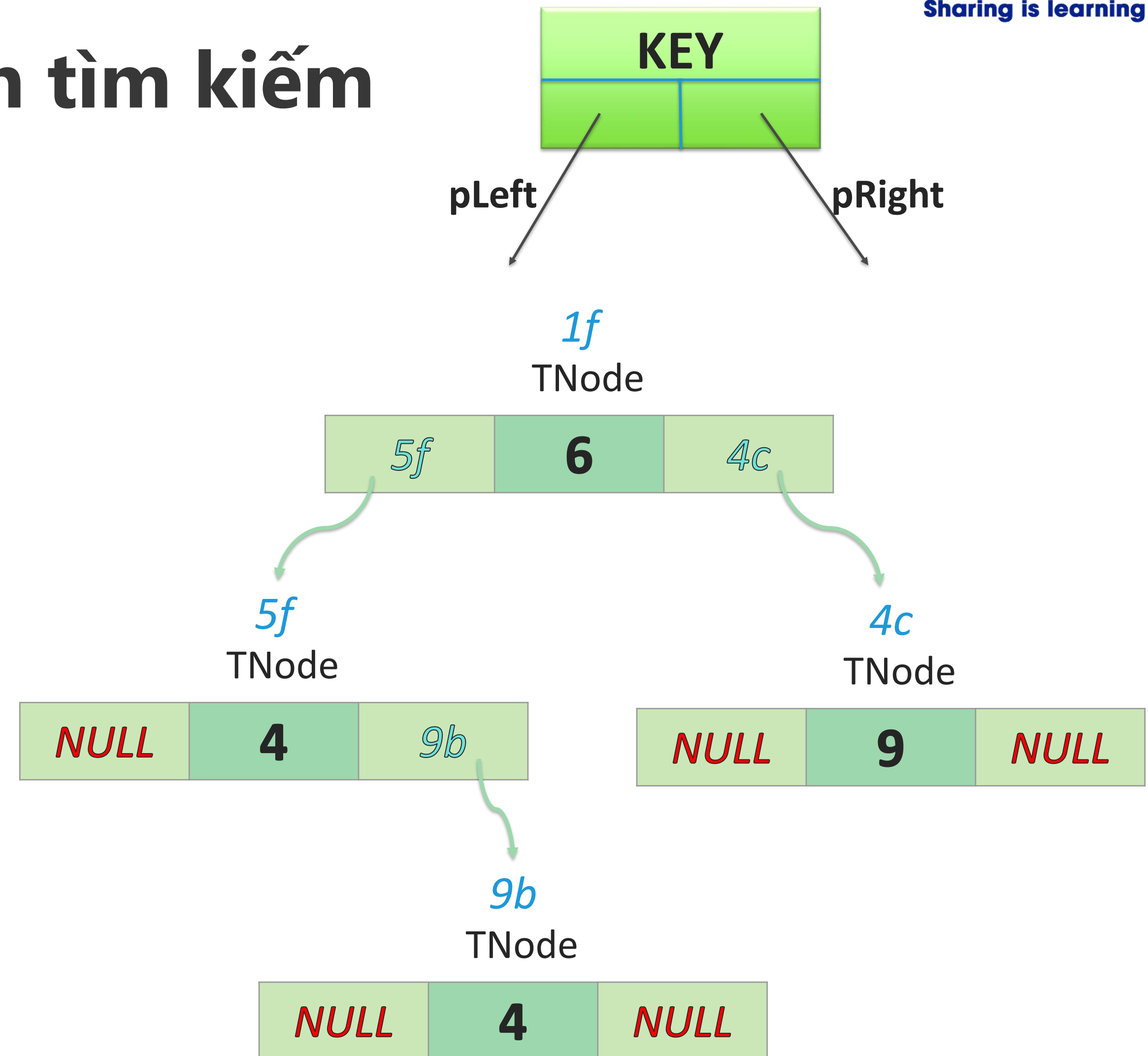
Sharing is learning

4. Các thao tác cây nhị phân tìm kiếm

Khai báo 1 nút (node) của cây:

```
struct TNode
{
    int Data;
    TNode* pLeft;
    TNode* pRight;
};

typedef TNode* Tree;
```



II. Cây nhị phân



Sharing is learning

4. Các thao tác Binary search tree

Tạo 1 nút

```
TNode* getTNode(int data)
{
    TNode* tmp = new TNode;
    tmp->Data = data;
    tmp->pLeft = NULL;
    tmp->pRight = NULL;
    return tmp;
}
```

Khởi tạo cây

```
void initTree(Tree& Root)
{
    Root = NULL;
}
```

II. Cây nhị phân



Sharing is learning

4. Các thao tác Binary search tree

Thao tác duyệt:

| | |
|---------------------------------|-----|
| Duyệt tiền thứ tự (duyệt trước) | NLR |
| | NRL |
| Duyệt trung thứ tự (duyệt giữa) | LNR |
| | RNL |
| Duyệt hậu thứ tự (duyệt sau) | LRN |
| | RLN |

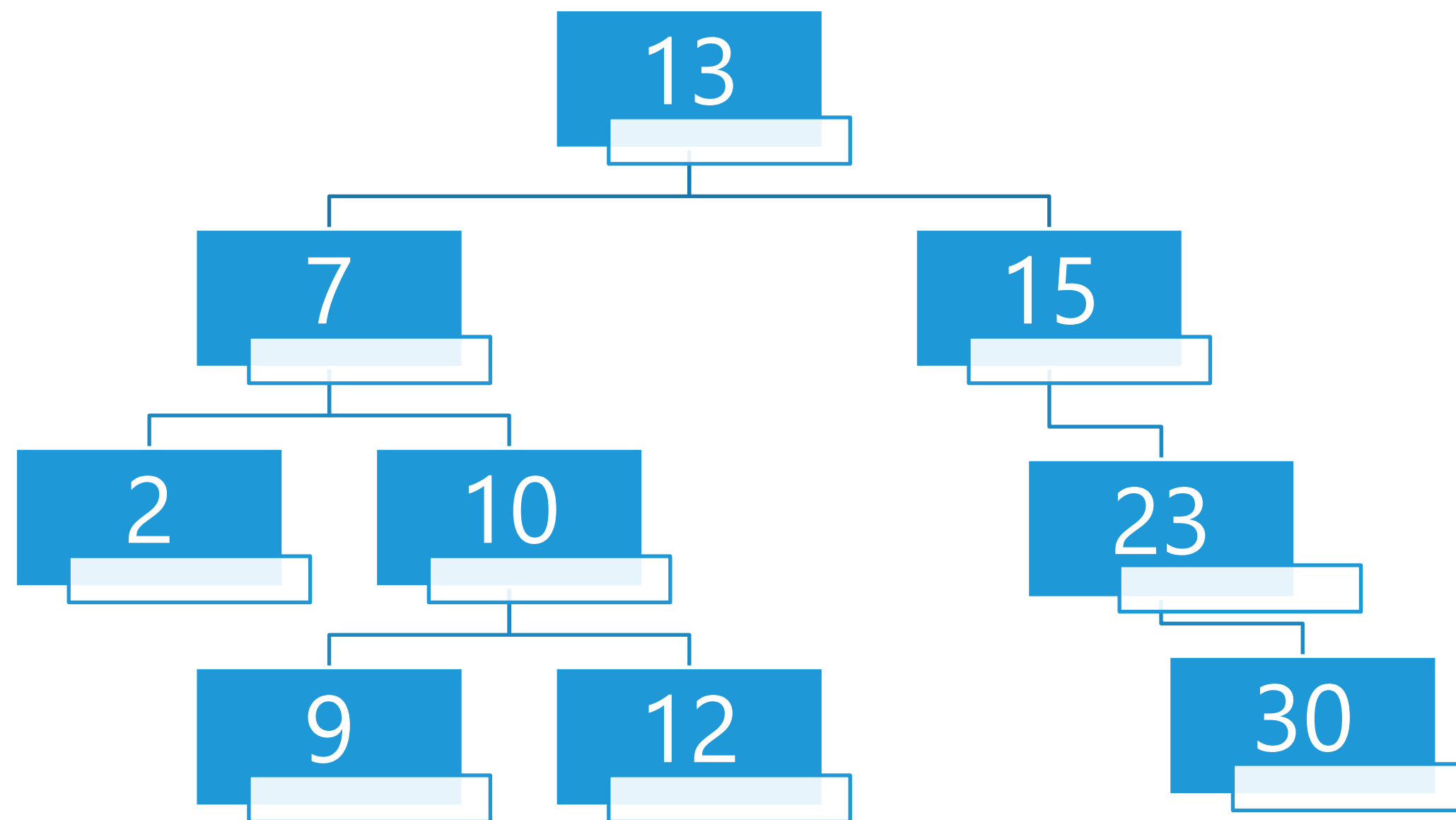
II. Cây nhị phân



Sharing is learning

4. Các thao tác Binary search tree

Thao tác duyệt:



- NLR: 13, 7, 2, 10, 9, 12, 15, 23, 30
- NRL: 13, 15, 23, 30, 7, 10, 12, 9, 2
- *LNR*: 2, 7, 9, 10, 12, 13, 15, 23, 30
- *RNL*: 30, 23, 15, 13, 12, 10, 9, 7, 2
- LRN: 2, 9, 12, 10, 7, 30, 23, 15, 13
- RLN: 30, 23, 15, 12, 9, 10, 2, 7, 13

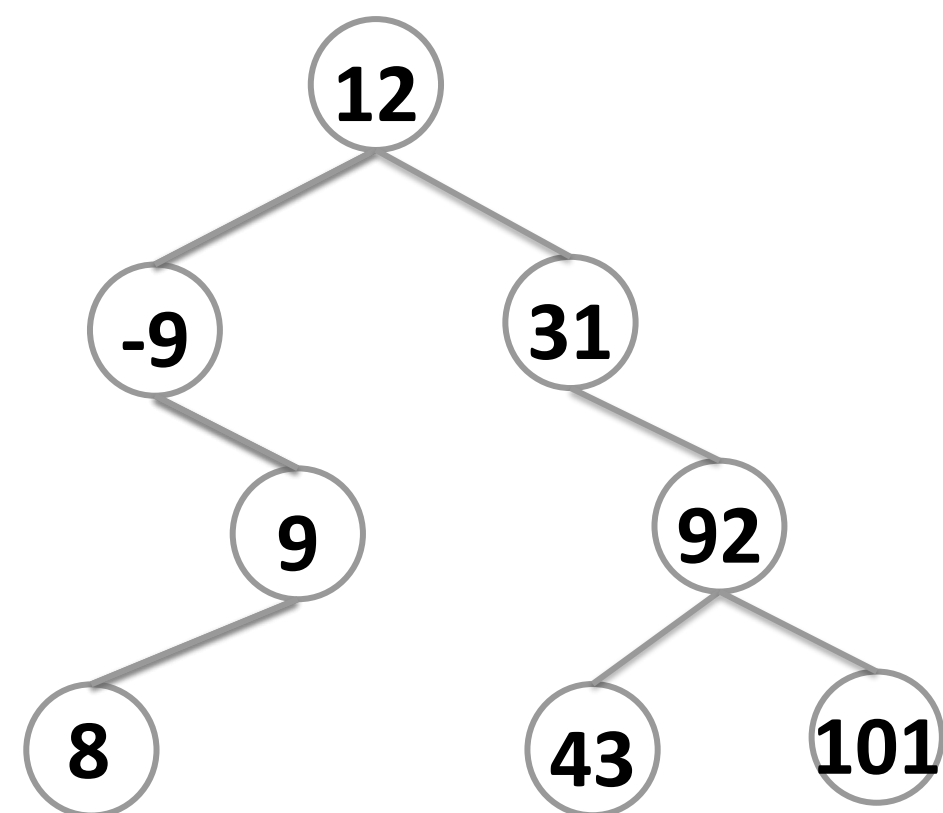
II. Cây nhị phân



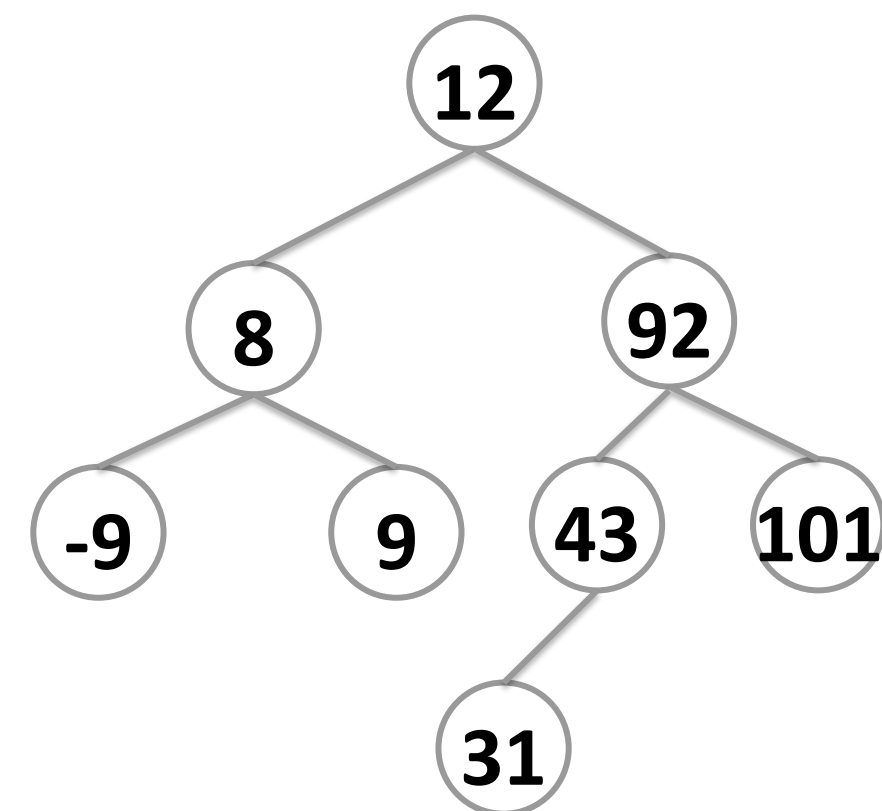
Sharing is learning

4. Các thao tác Binary search tree

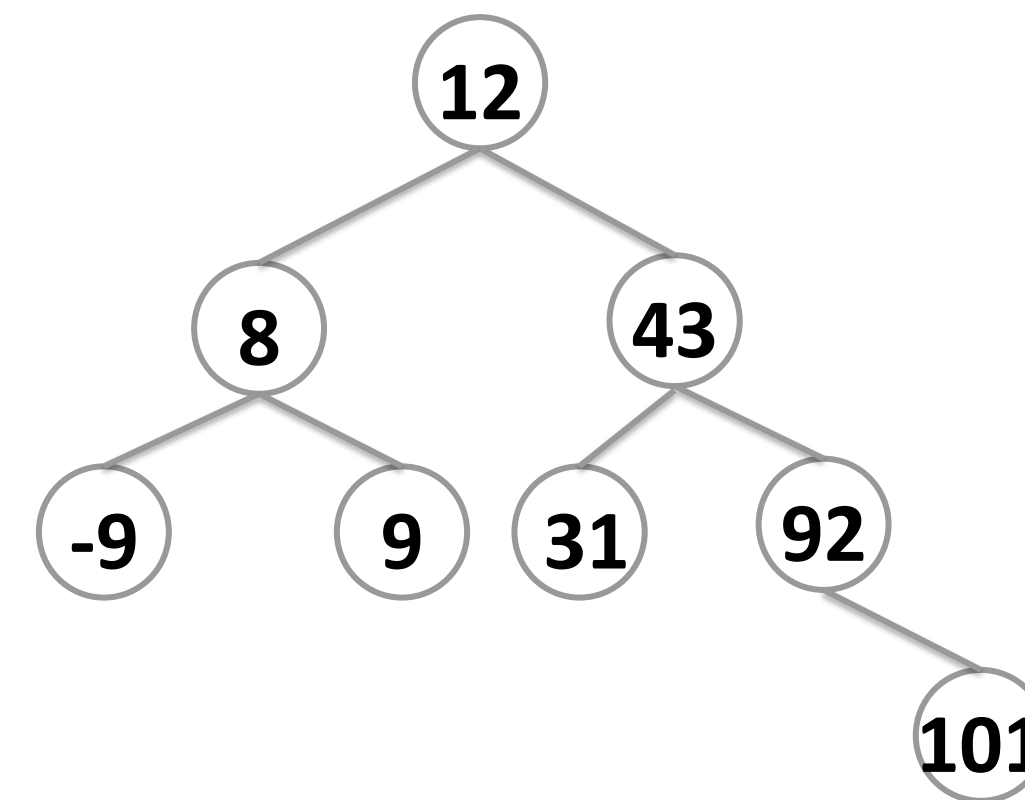
Câu 1: Cho dãy sau và tạo cây nhị phân tìm kiếm dựa trên thứ tự nhập vào của dãy: 12, 8, 31, 92, 43, 101, -9, 9
Chọn câu trả lời đúng:



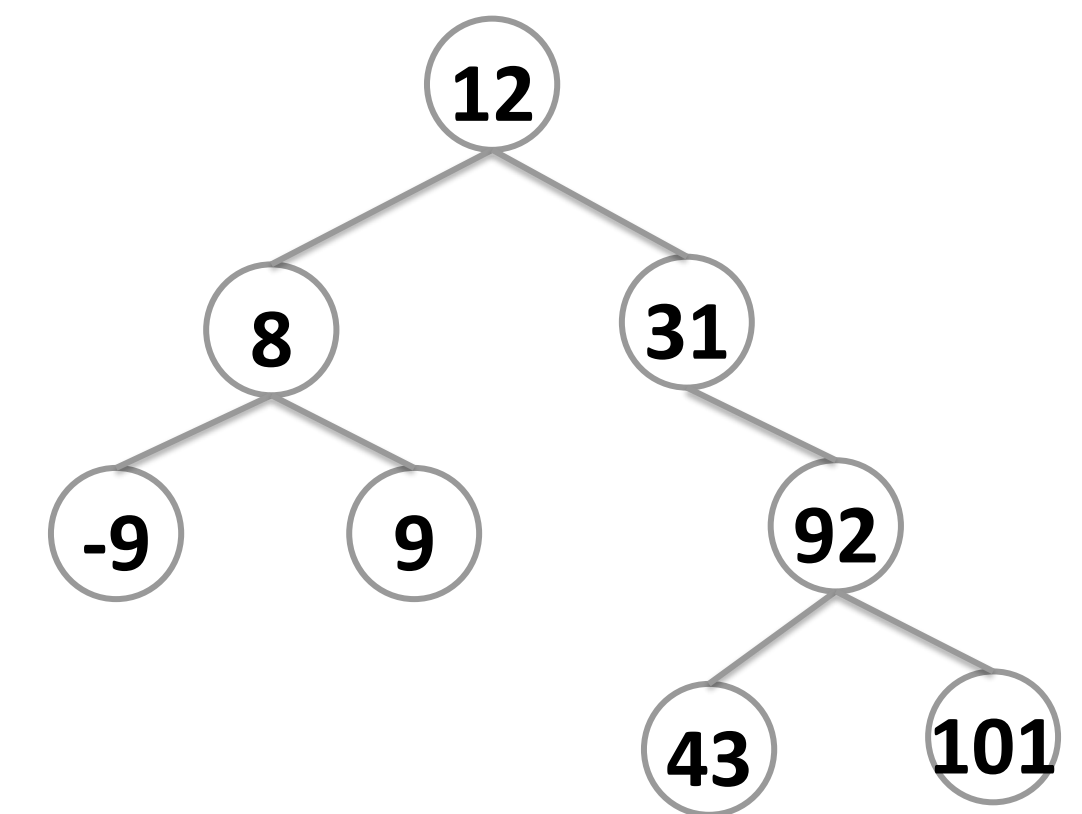
A



B



C



D

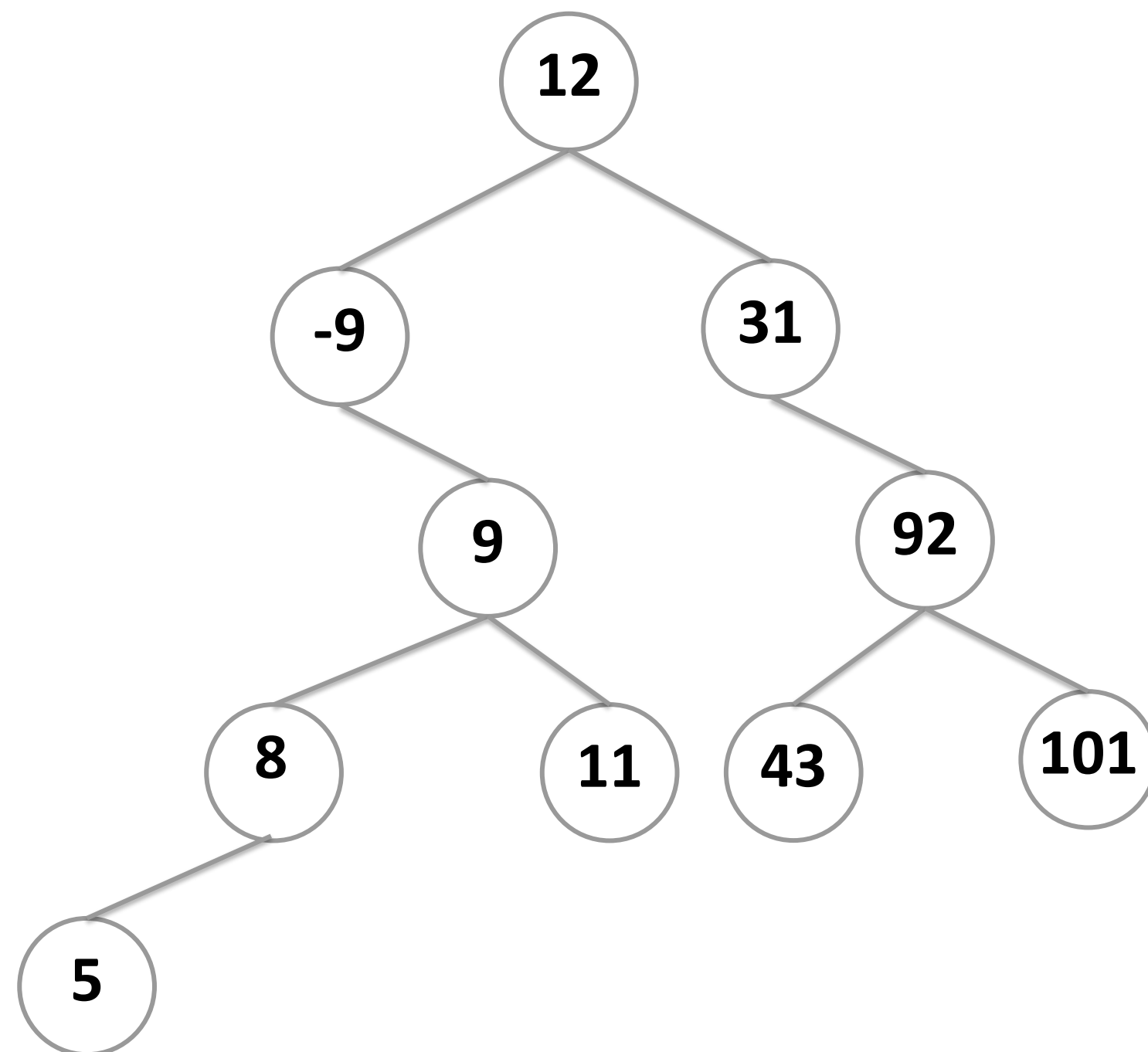
II. Cây nhị phân



Sharing is learning

4. Các thao tác Binary search tree

Câu 2: Cho cây nhị phân tìm kiếm sau và chọn phép duyệt đúng.



- A. LRN: 5, 8, 11, 9, -9, 43, 101, 31, 92, 12
- B. RLN: 101, 43, 92, 31, 11, 5, 8, -9, 9, 12
- C. NRL: 12, 31, 92, 101, 43, -9, 9, 11, 8, 5**
- D. LNR: -9, 5, 8, 9, 11, 12, 30, 43, 92, 101

II. Cây nhị phân



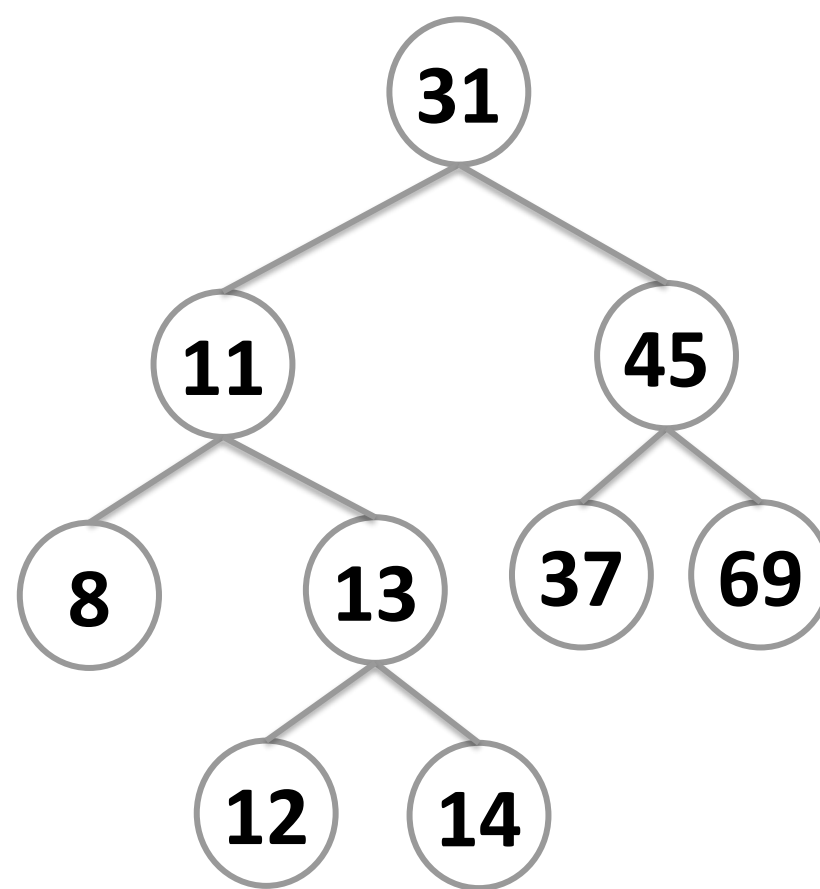
Sharing is learning

4. Các thao tác Binary search tree

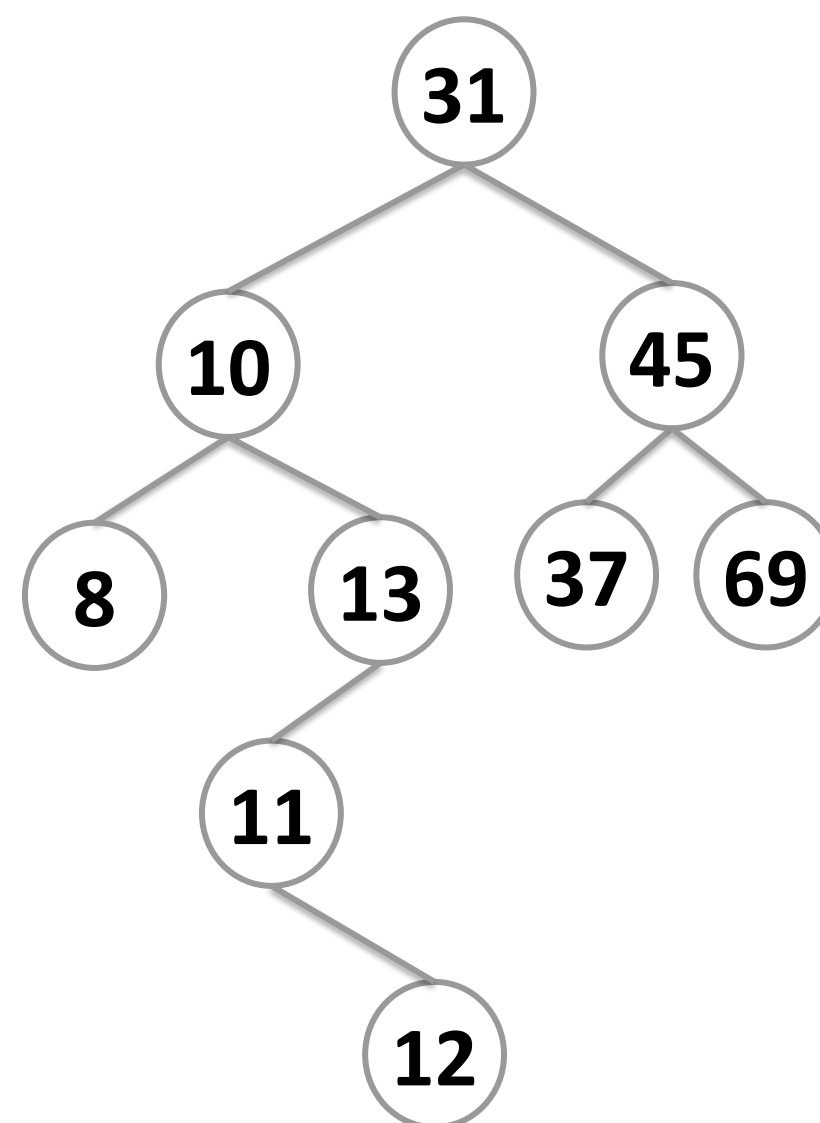
Câu 3: Cho kết quả phép duyệt trên cây nhị phân tìm kiếm.

Chọn cây nhị phân tìm kiếm thích hợp với phép duyệt đã cho.

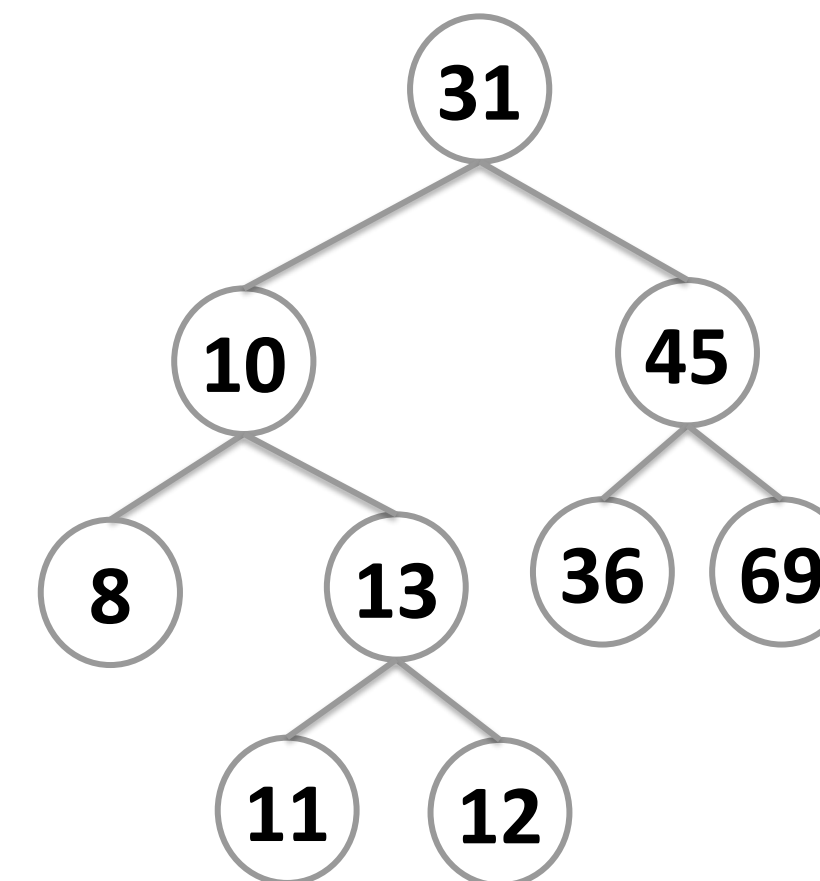
LRN: 8, 12, 11, 13, 10, 37, 69, 45, 31



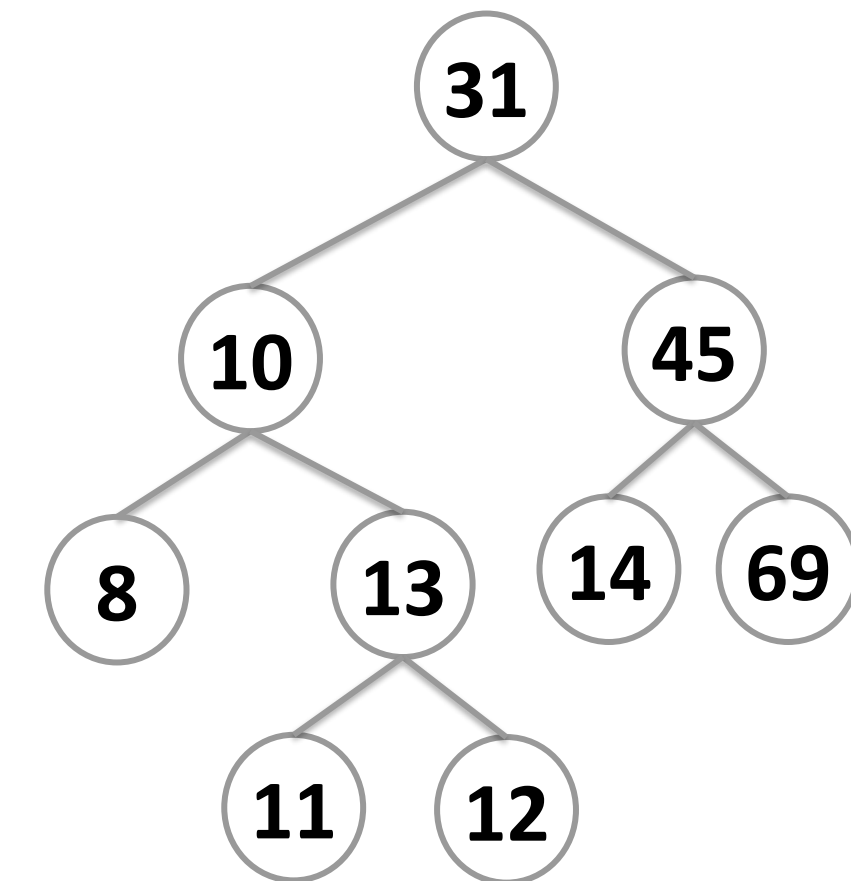
A



B



C



D

II. Cây nhị phân

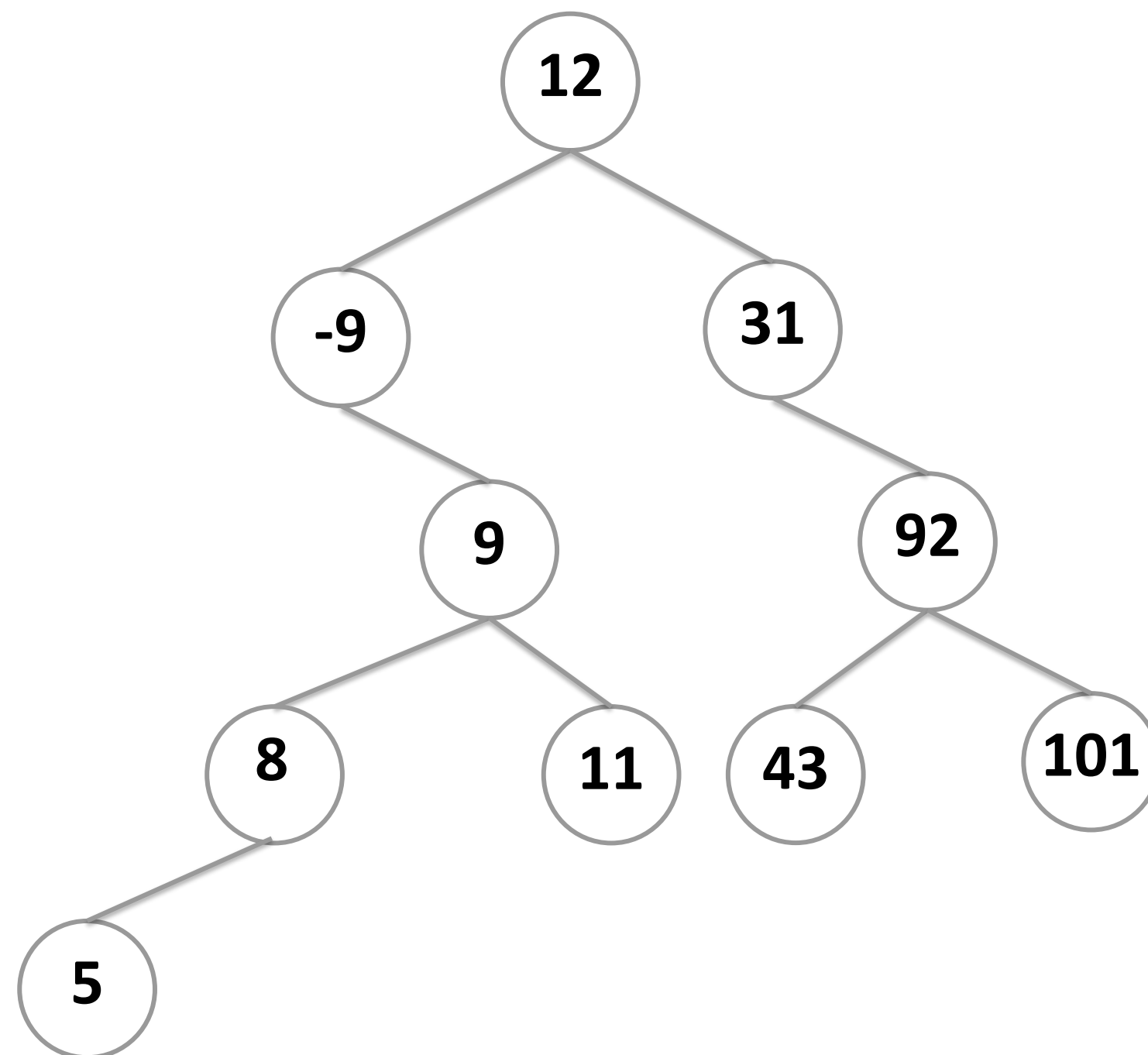


Sharing is learning

4. Các thao tác Binary search tree

Câu 4: Cho cây nhị phân tìm kiếm sau.

Liệt kê các node lá và node trong (node nhánh).



A. Node trong: -9, 31, 9, 92, 8
Node lá: 5, 8, 11, 43, 101

B. Node lá: 5, 43, 101, 11
Node trong: 12, -9, 31, 9, 92, 8

C. Node trong: -9, 31, 9, 92, 8
Node lá: 101, 11, 43, 5

D. Node trong: -9, 31, 9, 92, 8
Node lá: 101, 11, 42, 5

II. Cây nhị phân



Sharing is learning

4. Các thao tác Binary search tree

Thao tác duyệt: Duyệt trước

```
void NLR(Tree Root)
{
    if (Root != NULL)
    {
        /*<Xử lý Root>*/
        NLR(Root->pLeft);
        NLR(Root->pRight);
    }
}
```

```
void NRL(Tree Root)
{
    if (Root != NULL)
    {
        /*<Xử lý Root>*/
        NRL(Root->pRight);
        NRL(Root->pLeft);
    }
}
```


II. Cây nhị phân



Sharing is learning

4. Các thao tác Binary search tree

Thao tác duyệt: Duyệt giữa

```
void LNR(Tree Root)
{
    if (Root != NULL)
    {
        LNR(Root->pLeft);
        /*<Xử lý Root>*/
        LNR(Root->pRight);
    }
}
```

```
void RNL(Tree Root)
{
    if (Root != NULL)
    {
        RNL(Root->pRight);
        /*<Xử lý Root>*/
        RNL(Root->pLeft);
    }
}
```

II. Cây nhị phân



Sharing is learning

4. Các thao tác Binary search tree

Thao tác duyệt: Duyệt sau

```
void LRN(Tree Root)
{
    if (Root != NULL)
    {
        LRN(Root->pLeft);
        LRN(Root->pRight);
        /*<Xử lý Root>*/
    }
}
```

```
void RLN(Tree Root)
{
    if (Root != NULL)
    {
        RLN(Root->pRight);
        RLN(Root->pLeft);
        /*<Xử lý Root>*/
    }
}
```

II. Cây nhị phân



Sharing is learning

4. Các thao tác Binary search tree

CHÈN NÚT VÀO CÂY:

➤ Nếu **Root khác rỗng**:

- Nếu x **lớn hơn** giá trị Root:
 - **Đi về cây con bên phải** Root.
- Nếu x **nhỏ hơn** giá trị Root:
 - **Đi về cây con bên trái** Root.
- Còn nếu x **bằng** giá trị của Root:
 - **Không làm gì cả!**

➤ Ngược lại (**Root rỗng**):

- Cấp phát Root.
- Gán giá trị Root bằng x.
- Gán con trỏ pLeft, pRight bằng NULL.

II. Cây nhị phân



Sharing is learning

4. Các thao tác Binary search tree

CHÈN NÚT VÀO CÂY:

```
int insertTNode(Tree& Root, int data)
{
    if (Root)
    {
        if (data > Root->Data)
            return insertTNode(Root->pRight, data);
        else if (data < Root->Data)
            return insertTNode(Root->pLeft, data);
        else
            return 0;
    }
}
```

```
.....
else
{
    Root = new TNode;
    if (Root == NULL)
        return -1;
    Root->Data = data;
    Root->pLeft = NULL;
    Root->pRight = NULL;
    return 1;
}
```

```
}
```

II. Cây nhị phân



Sharing is learning

4. Các thao tác Binary search tree

CHÈN NÚT VÀO CÂY:

```
void insertTNode(Tree& Root, int& value) {  
    if (!Root) {  
        Root = new TNode;  
        Root->Data = value;  
        Root->pLeft = Root->pRight = NULL;  
    }  
    else {  
        if (Root->Data > value) {  
            insertTNode(Root->pLeft, value);  
        }  
        else if (Root->Data < value) {  
            insertTNode(Root->pRight, value);  
        }  
    }  
}
```


II. Cây nhị phân



Sharing is learning

4. Các thao tác Binary search tree

Tìm nút trên cây:

```
TNode* searchTNode(Tree Root, int data)
{
    TNode* tmp = Root;
    while (tmp != NULL)
    {
        if (tmp->Data > data)
            tmp = tmp->pRight;
        else if (tmp->Data < data)
            tmp = tmp->pLeft;
        else
            return tmp;
    }
    return NULL;
}
```

Không đệ qui

```
TNode* searchTNode(Tree Root, int data)
{
    if (!Root)
        return NULL;
    if (Root->Data > data)
        return searchTNode(Root->pLeft, data);
    else if (Root->Data < data)
        return searchTNode(Root->pRight, data);
    else
        return Root;
}
```

Đệ qui

II. Cây nhị phân



Sharing is learning

4. Các thao tác Binary search tree

Xóa nút trên cây:

Có 3 TH:

- Nút đó **là lá** (không có nút con).
- Nút đó **có 1 nút con**.
- Nút đó **có 2 nút con**:
 - ✓ **Tìm nút thay thế** để luôn đảm bảo qui tắc cây nhị phân:
 - Tìm nút **trái nhất** của **cây con bên phải**.
 - Tìm nút **phải nhất** của **cây con bên trái**.
 - ✓ **Gán giá trị** nút thế mạng cho nút đang cần xóa.
 - ✓ **Xóa nút thế mạng**: Lúc này nút thế mạng rơi vào 2 trường hợp trên 1, 2 và nếu rơi vào trường hợp 2 thì chắc chắn chỉ có nút con bên phải.

II. Cây nhị phân



Sharing is learning

4. Các thao tác Binary search tree

Xóa nút trên cây:

```
void deleteKey(Tree& key, int& x) {
    if (key != NULL) {
        if (key->Data == x) {
            TNode* tmp = key;
            if (key->pLeft == NULL) {
                key = key->pRight;
                delete tmp;
            }
            else if (key->pRight == NULL) {
                key = key->pLeft;
                delete tmp;
            }
            else {
                replace(key, key->pRight);
            }
        }
        else {
            if (key->Data > x) {
                deleteKey(key->pLeft, x);
            }
            else {
                deleteKey(key->pRight, x);
            }
        }
    }
}
```

```
void replace(Tree& T, Tree& key) {
    if (key != NULL) {
        if (key->pLeft == NULL) {
            T->Data = key->Data;
            TNode* tmp = key;
            key = key->pRight;
            tmp->pRight = NULL;
            delete tmp;
        }
        else {
            replace(T, key->pLeft);
        }
    }
}
```

II. Cây nhị phân



Sharing is learning

4. Các thao tác Binary search tree

Chiều cao cây:

```
int DepthTree(Tree T)
{
    if (T == NULL)
        return 0;
    else
        return max(DepthTree(T->pLeft), DepthTree(T->pRight)) + 1;
}

int HeightTree(Tree T)
{
    return DepthTree(T) - 1;
}
```

Nội dung Training

I. Các thuật toán sắp xếp

II. Cây nhị phân

III. Bảng băm



Sharing is learning



Sharing is learning



III. Bảng băm (Hash table)

1. Băm.

2. Xử lí đụng độ.



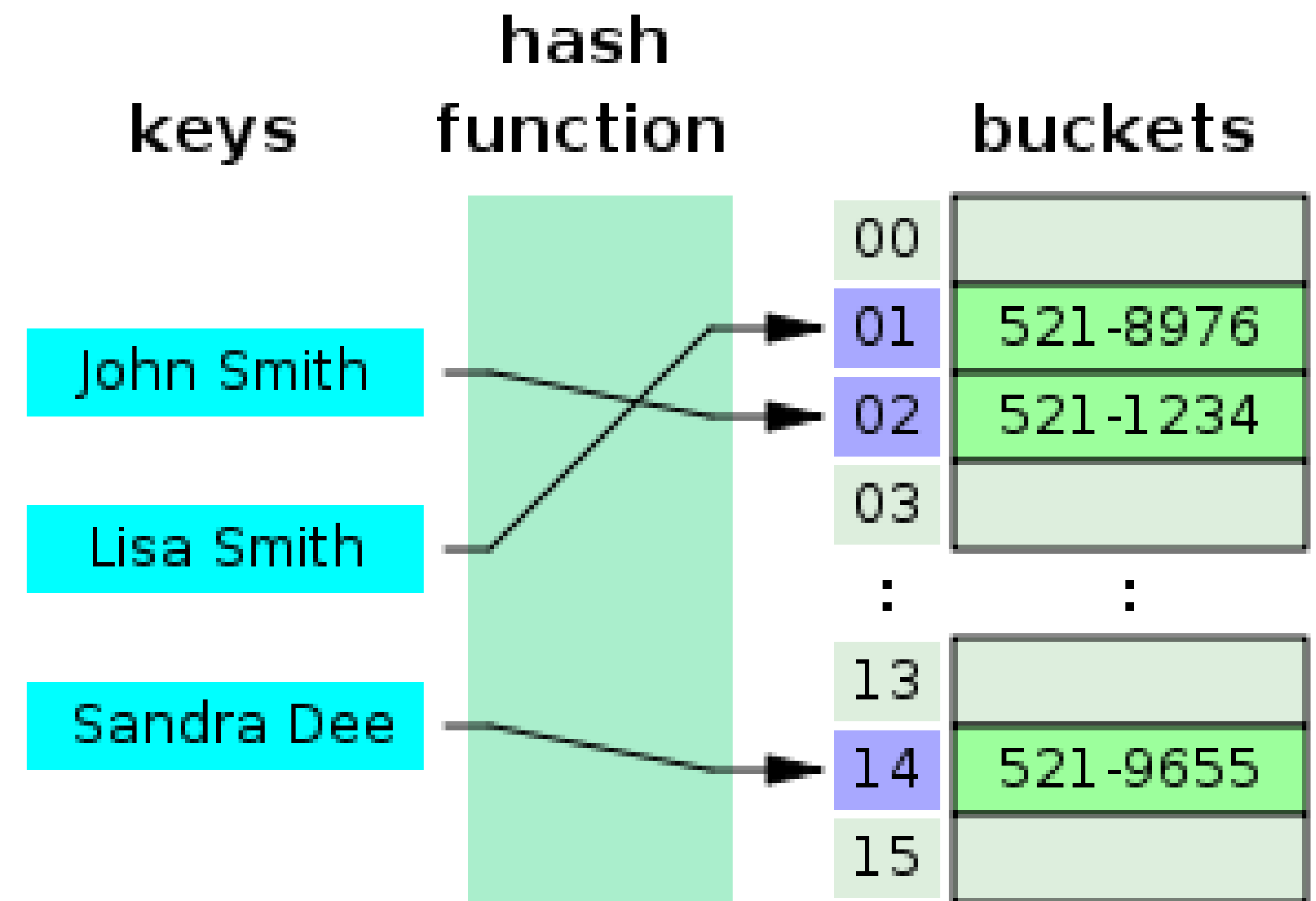
III. Bảng băm



Sharing is learning

1. Băm

- Khái niệm bảng băm:
 - Bảng băm là một CTDL trong đó mỗi phần tử là một cặp (khóa, giá trị)



III. Bảng băm



1. Băm

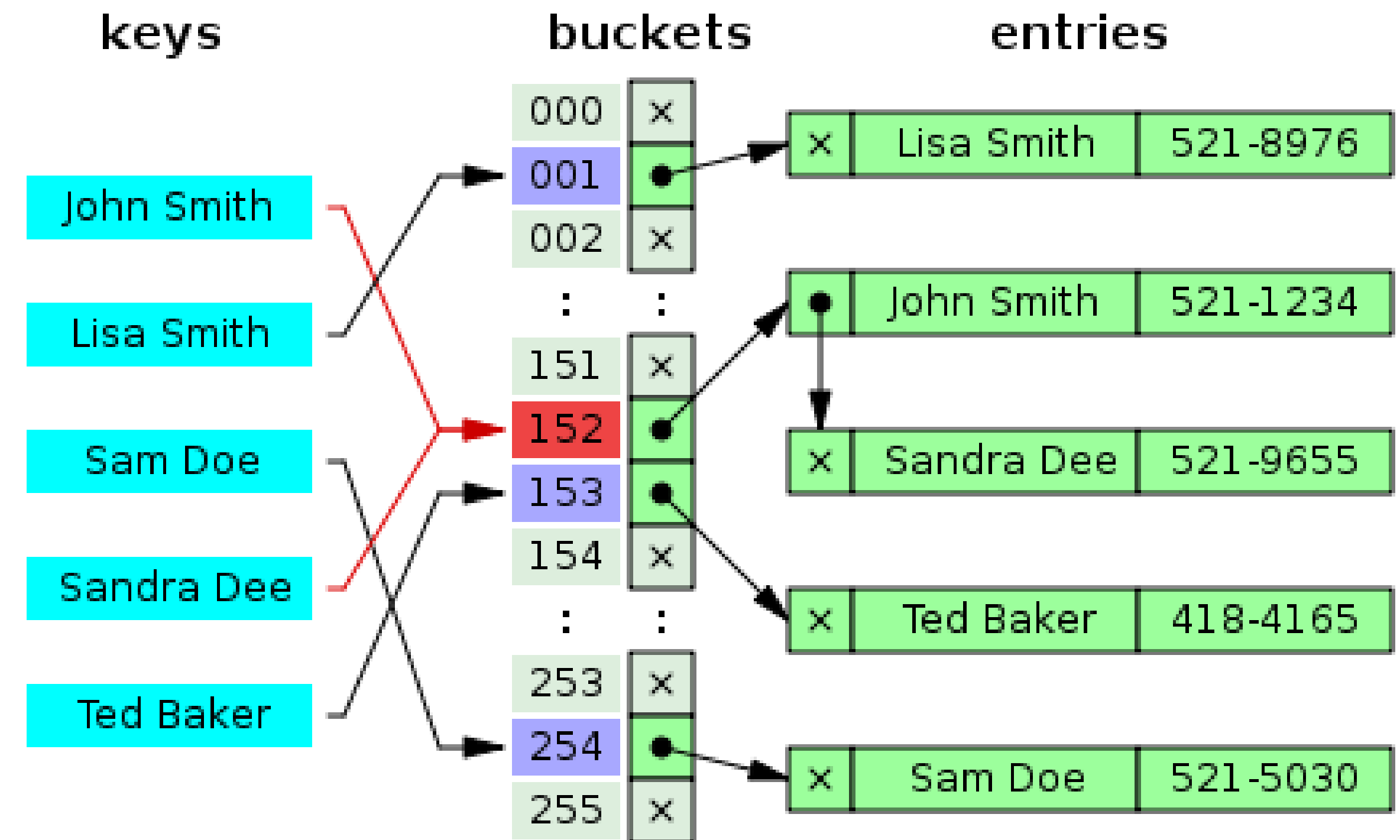
- Hàm băm là một ánh xạ biến đổi khóa thành chỉ số của mảng
- Hàm băm đủ tốt:
 - Tính toán **nhANH** (không phải là thuật toán)
 - Các **khóa được phân bố đều** trong bảng
 - **Ít** xảy ra đụng độ
 - Giải quyết vấn đề băm với các **khóa không là số nguyên**

III. Bảng băm



2. Xử lí đụng độ

- Khái niệm sự đụng độ:
- Hiện tượng các khóa khác nhau nhưng băm cùng địa chỉ như nhau



III. Bảng băm



Sharing is learning

2. Xử lí đụng độ

- Hai phương pháp:
 - Phương pháp nối kết.
 - Phương pháp băm lại:
 - Dò tuyến tính.
 - Dò bậc 2.
 - Băm kép.



Sharing is learning

III. Bảng băm



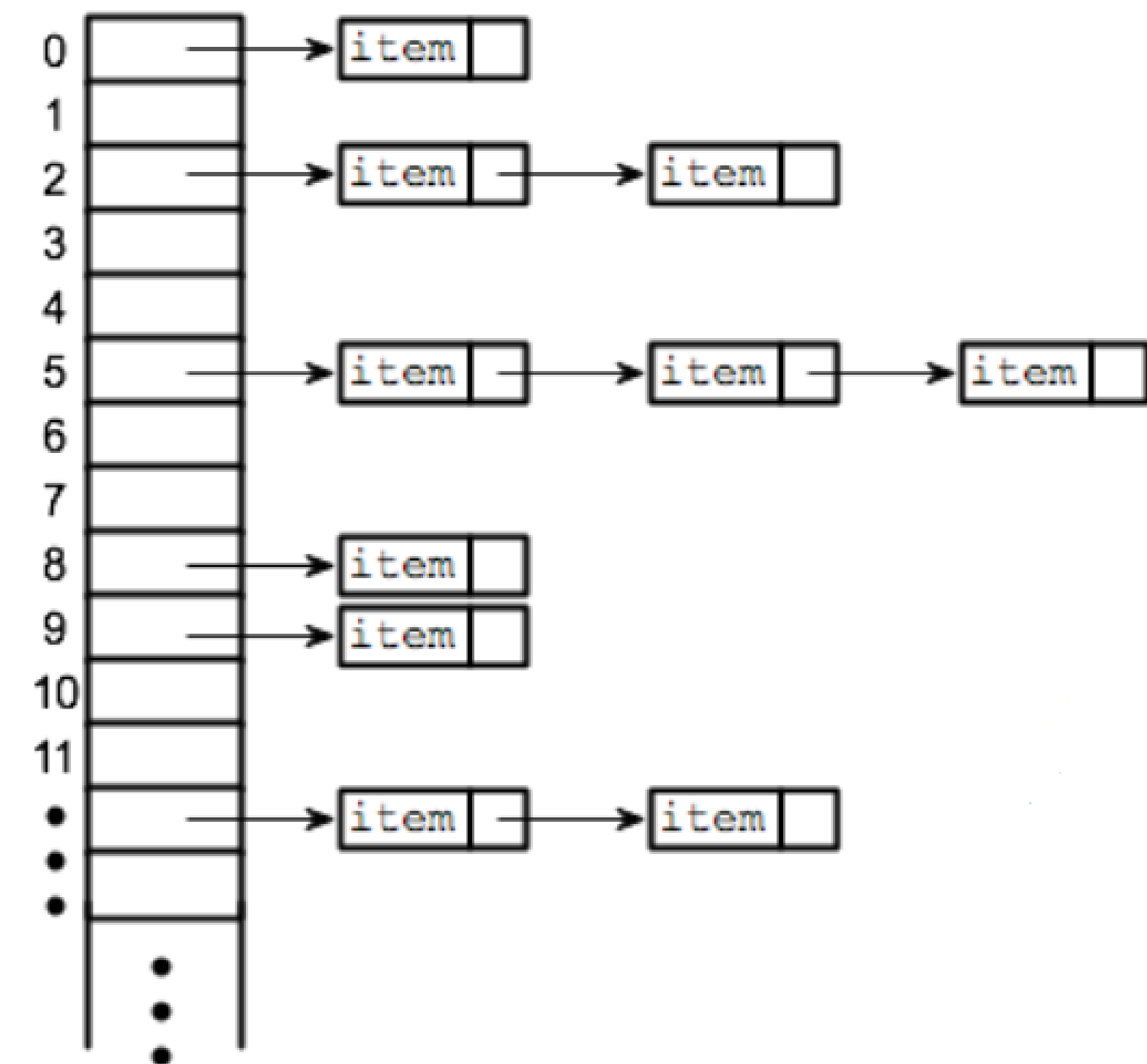
Sharing is learning

2. Xử lý đụng độ

Phương pháp nối kết:

Nối kết trực tiếp

- Chèn một phần tử vào bảng băm, ta chèn vào một danh sách liên kết.
- Khi xảy ra đụng độ, ta chèn chúng vào chung một danh sách liên kết.



III. Bảng băm

2. Xử lý đụng độ

Phương pháp nối kết:

Nối kết hợp nhất

- Chèn một phần tử vào bảng băm, ta chèn phần tử đó vào với địa chỉ liên kết -1.
- Khi xảy ra đụng độ, ta tìm ngược từ cuối bảng lên đầu bảng vị trí còn trống để chèn. Cập nhật trường next sao cho các nút xung đột được nối với nhau như 1 danh sách liên kết

22, 32, 43, 23, 51, 42

| | KEY | NEXT |
|---|------|------|
| 0 | NULL | -1 |
| 1 | 51 | -1 |
| 2 | 22 | 6 |
| 3 | 43 | 5 |
| 4 | 42 | -1 |
| 5 | 23 | -1 |
| 6 | 32 | 4 |

$$f(\text{key}) = \text{key} \% 5$$

III. Bảng băm



Sharing is learning

2. Xử lý đụng độ

Phương pháp băm lại:

- Dò tuyến tính:

$$H(\text{key}, i) = (H_1(\text{key}) + i) \% M$$

- Dò bậc 2:

$$H(\text{key}, i) = (H_1(\text{key}) + i^2) \% M$$

- Băm kép:

H_1, H_2 .

$$H(\text{key}, i) = (H_1(\text{key}) + i * H_2(\text{key})) \% M$$

M: Kích thước mảng hoặc 1 số nào đó

i: Lần đụng độ

$H_1(\text{key})$

$H_2(\text{key})$

: Hàm băm bất kỳ

Nội dung Training

I. Các thuật toán sắp xếp

II. Cây nhị phân

III. Bảng băm

IV. Đồ thị



Sharing is learning



Sharing is learning

IV. Đồ thị



Sharing is learning

IV. Đồ thị (Graph)

1. Đồ thị IVà các khái niệm trên đồ thị
2. Biểu diễn đồ thị trên máy tính
3. Duyệt đồ thị theo chiều sâu IVà chiều rộng



Sharing is learning

IV. Đồ thị



Sharing is learning

1. Đồ thị và các khái niệm trên đồ thị

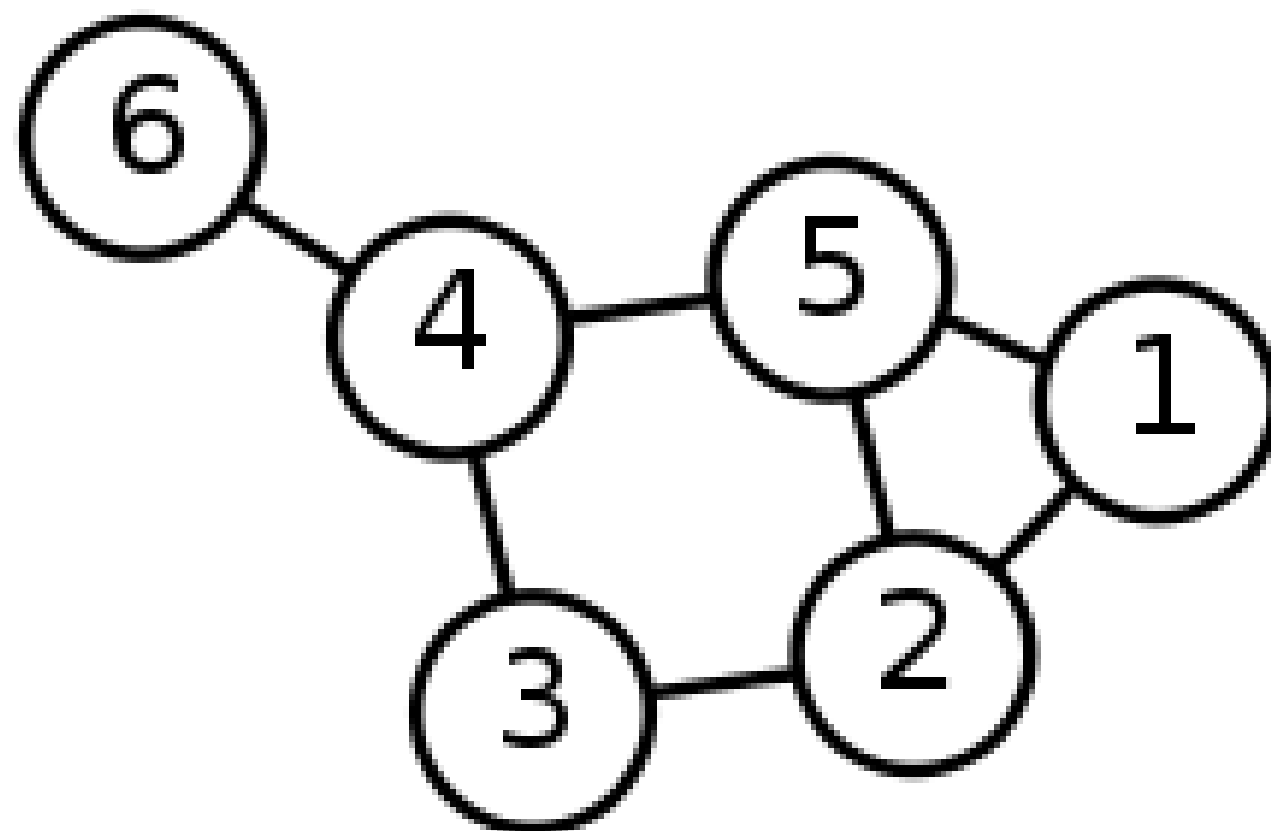
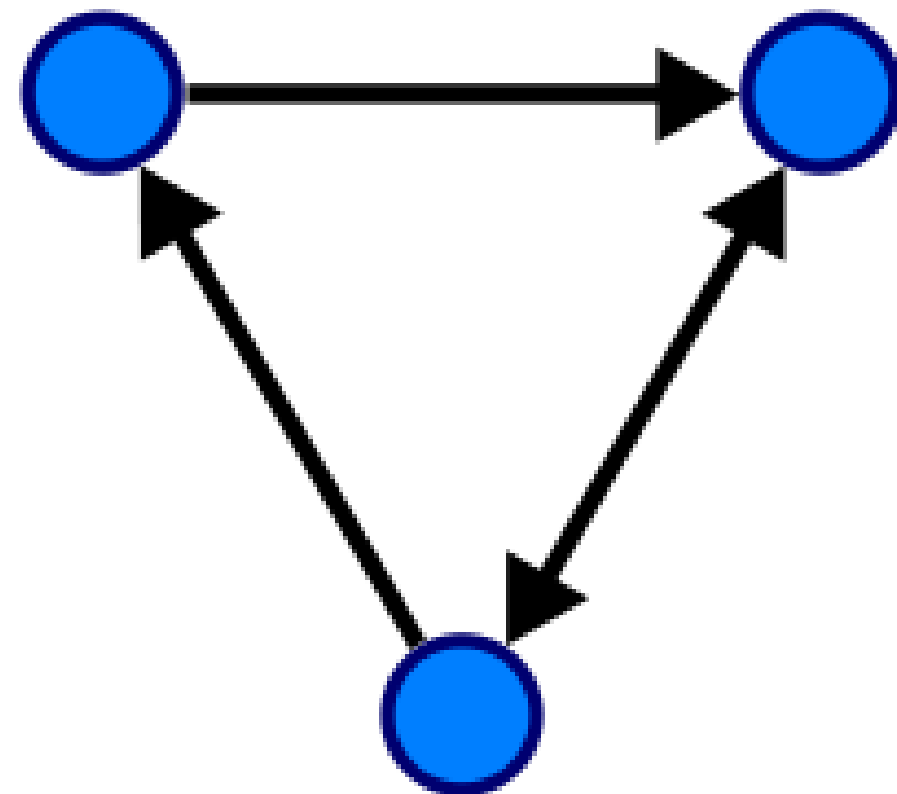
- Trong **toán rời rạc**: là cấu trúc rời rạc **có tính trực quan cao**, được sử dụng để **biểu diễn một tập đối tượng có quan hệ với nhau** theo một cách nào đó
- Định nghĩa hình thức: đồ thị G được xác định bởi một cặp (V, E) , trong đó:
 - V là tập đỉnh
 - E là tập các cạnh nối cặp đỉnh $E \subseteq \{(u,v) \mid u, v \in V\}$

IV. Đồ thị

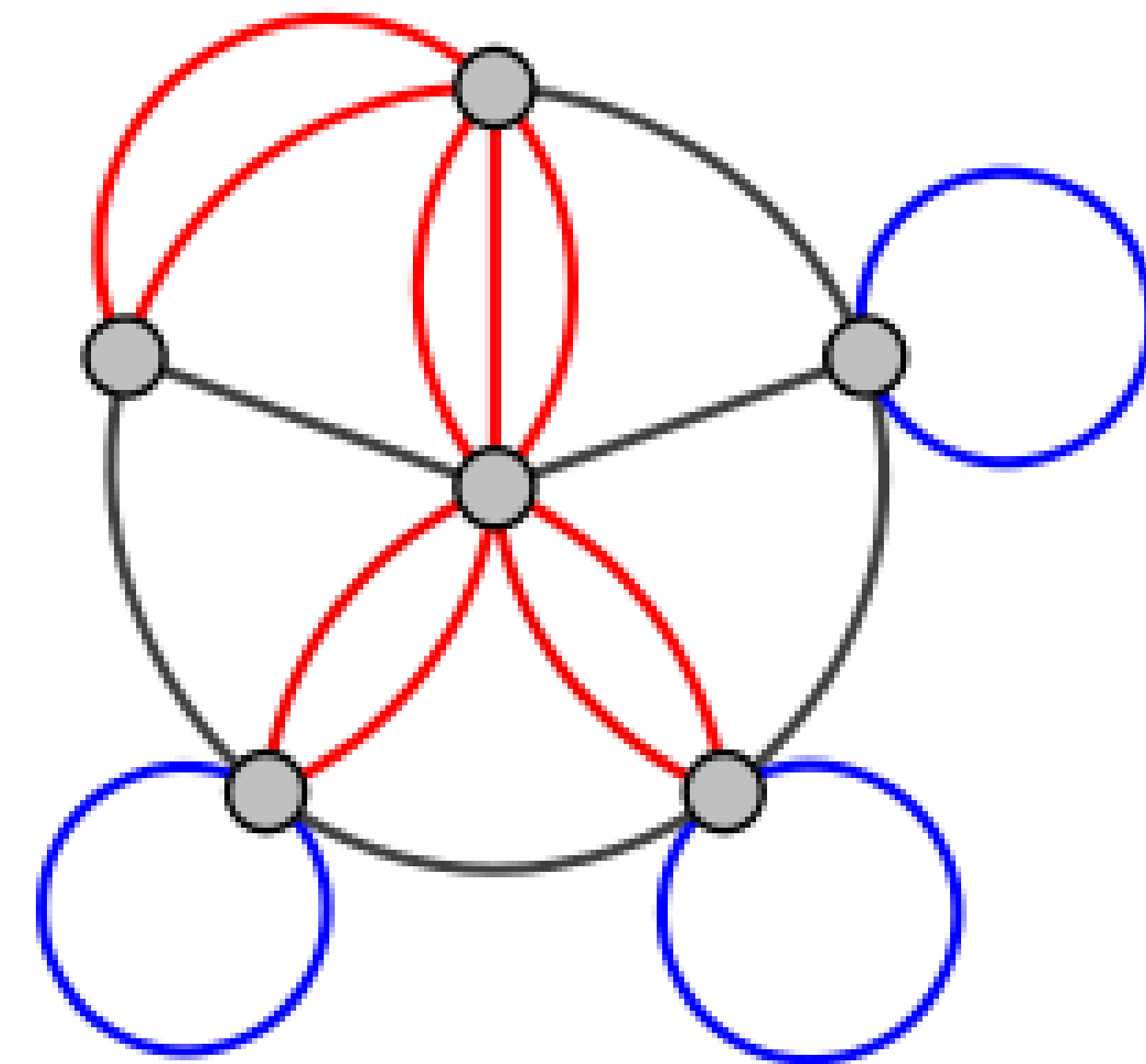


Sharing is learning

Đơn đồ thị - Simple Graph



Đa đồ thị - Multi-Graph



IV. Đồ thị



Sharing is learning

2. Biểu diễn đồ thị trên máy tính

- Ma trận kề
- Danh sách kề
- Danh sách cạnh (cung)
- Ma trận liên thuộc Đỉnh – Cạnh

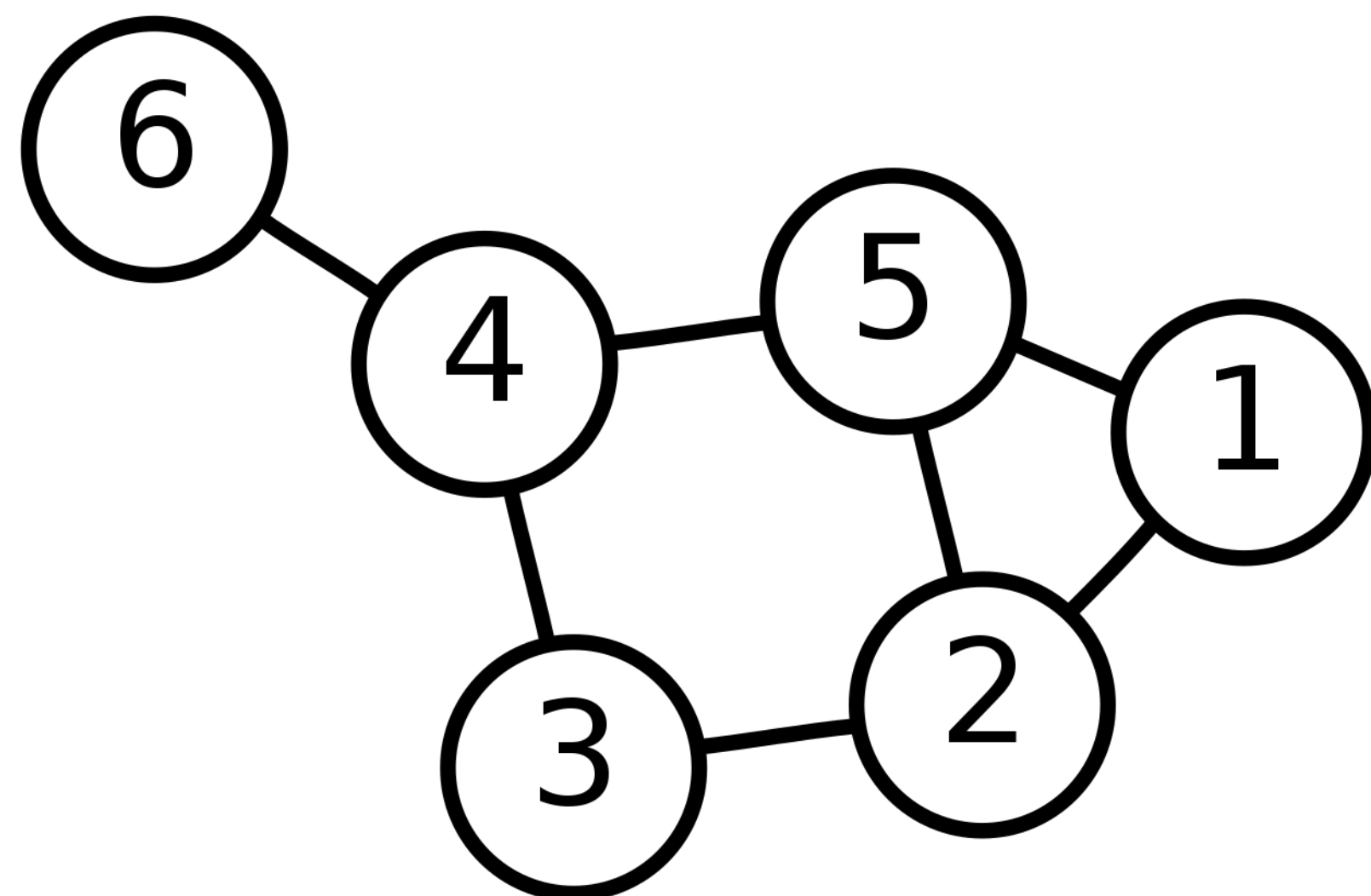
IV. Đồ thị



2. Biểu diễn đồ thị trên máy tính

Ma trận kề (Ma trận liên kết) – Adjacency Matrix:

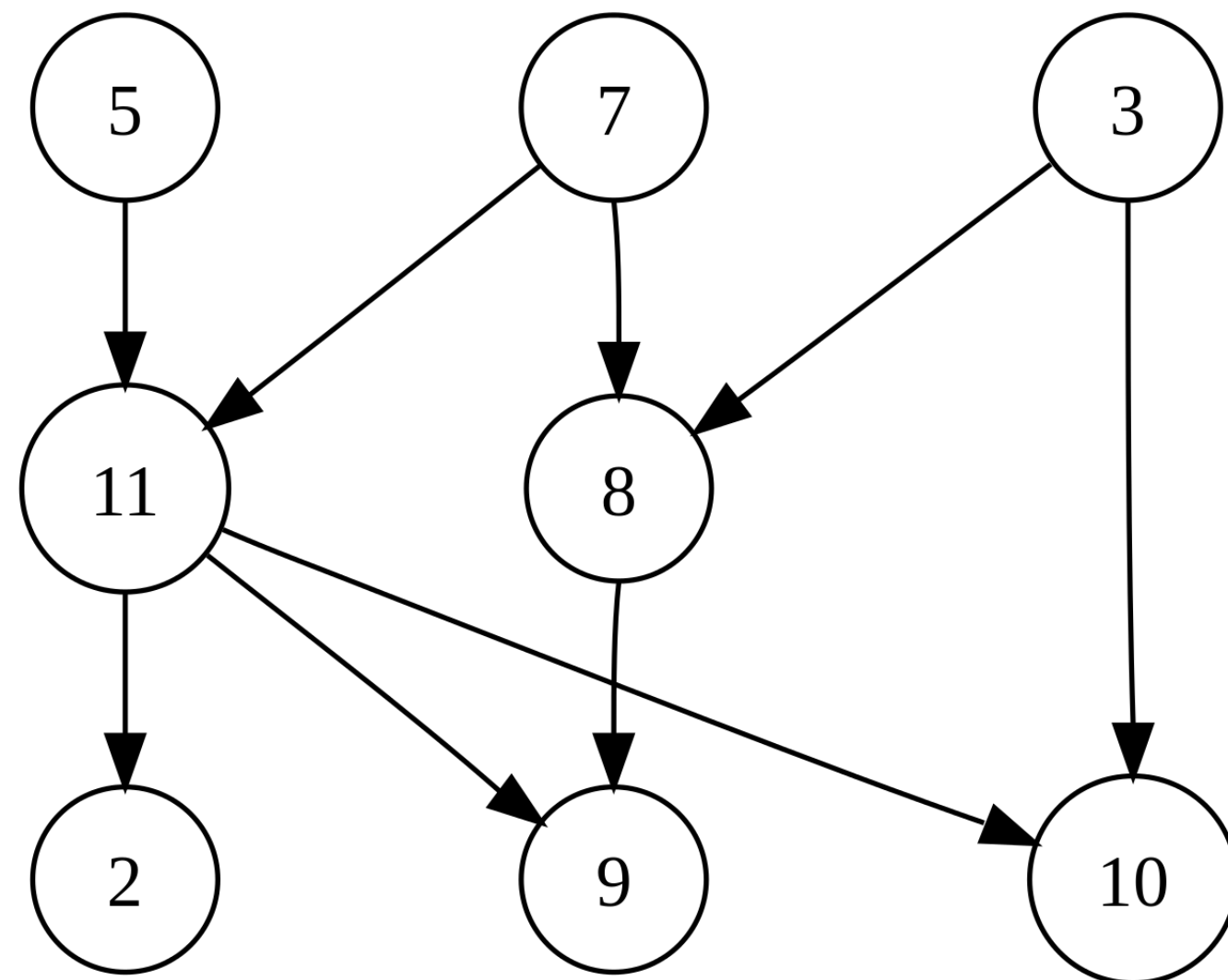
- Ma trận vuông cấp n , với n là số đỉnh.
- Thể hiện liên kết giữa các đỉnh với nhau. (Thể hiện số cạnh nối giữa các đỉnh)
- Quy tắc: Xét từng phần tử theo từng dòng.
- Bậc của đỉnh là tổng các phần tử theo dòng.
- **Tổng** các phần tử **theo dòng** là **bậc ngoài** (out-degree) của 1 đỉnh.
- **Tổng** các phần tử **theo cột** là **bậc trong** (in-degree) của 1 đỉnh.



| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | | 1 | | | 1 | |
| 2 | 1 | | 1 | | 1 | |
| 3 | | 1 | | 1 | | |
| 4 | | | 1 | | 1 | 1 |
| 5 | 1 | 1 | | 1 | | |
| 6 | | | | 1 | | |

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 2 | 1 | 0 | 1 | 0 | 1 | 0 |
| 3 | 0 | 1 | 0 | 1 | 0 | 0 |
| 4 | 0 | 0 | 1 | 0 | 1 | 1 |
| 5 | 1 | 1 | 0 | 1 | 0 | 0 |
| 6 | 0 | 0 | 0 | 1 | 0 | 0 |

2. Biểu diễn đồ thị trên máy tính



| | 2 | 3 | 5 | 7 | 8 | 9 | 10 | 11 |
|----|---|---|---|---|---|---|----|----|
| 2 | | | | | | | | |
| 3 | | | | | 1 | | 1 | |
| 5 | | | | | | | | 1 |
| 7 | | | | | 1 | | | 1 |
| 8 | | | | | | 1 | | |
| 9 | | | | | | | | |
| 10 | | | | | | | | |
| 11 | 1 | | | | | 1 | 1 | |

IV. Đồ thị



Sharing is learning

2. Biểu diễn đồ thị trên máy tính

Danh sách kề – Adjacency List:

- Một mảng n con trỏ, mỗi con trỏ quản lí 1 list (n là số đỉnh).
- Mỗi 1 phần tử trong mảng đại diện cho 1 đỉnh.
- List dùng để lưu trữ các đỉnh kề đỉnh với đỉnh đang xét.



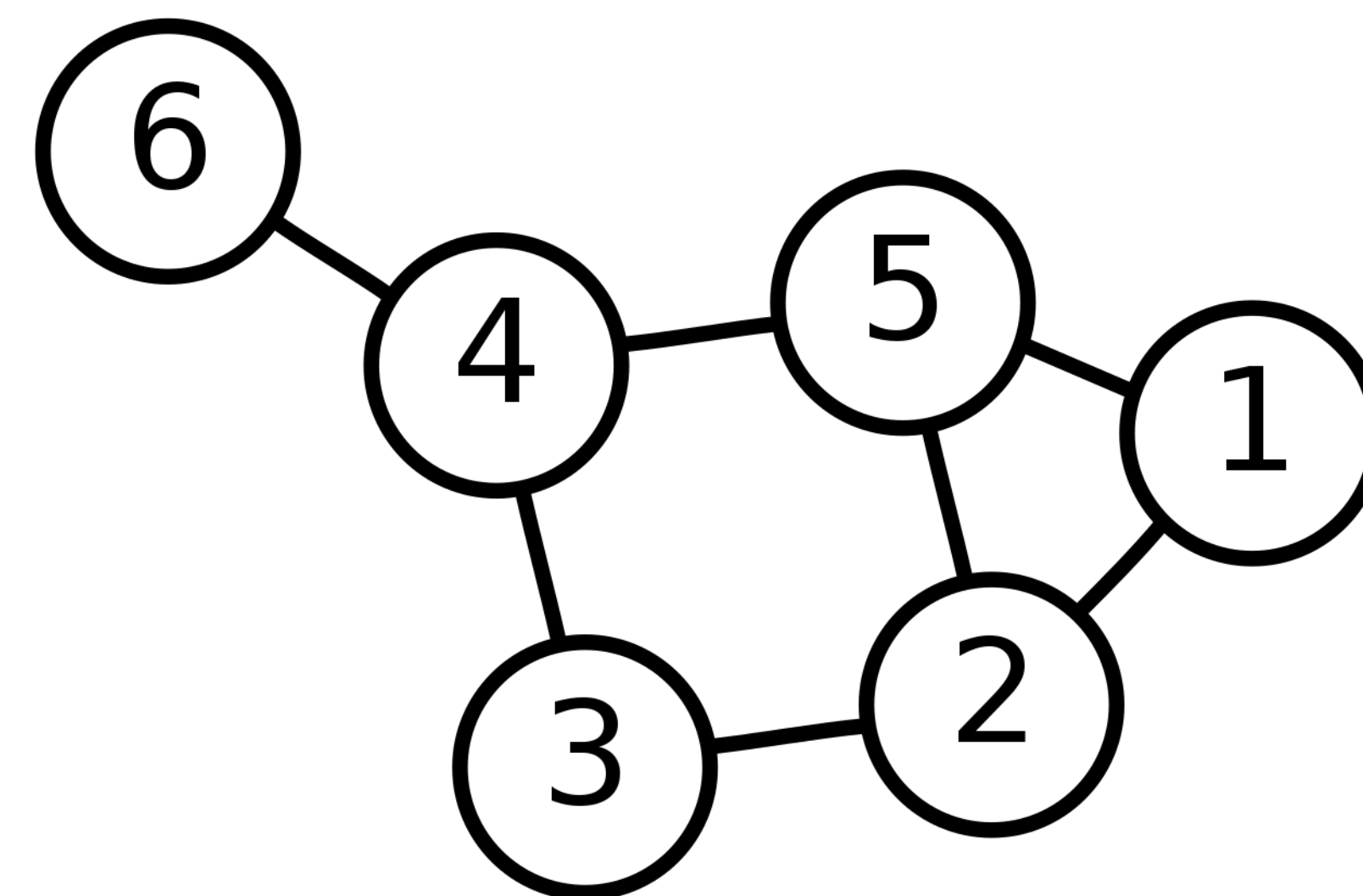
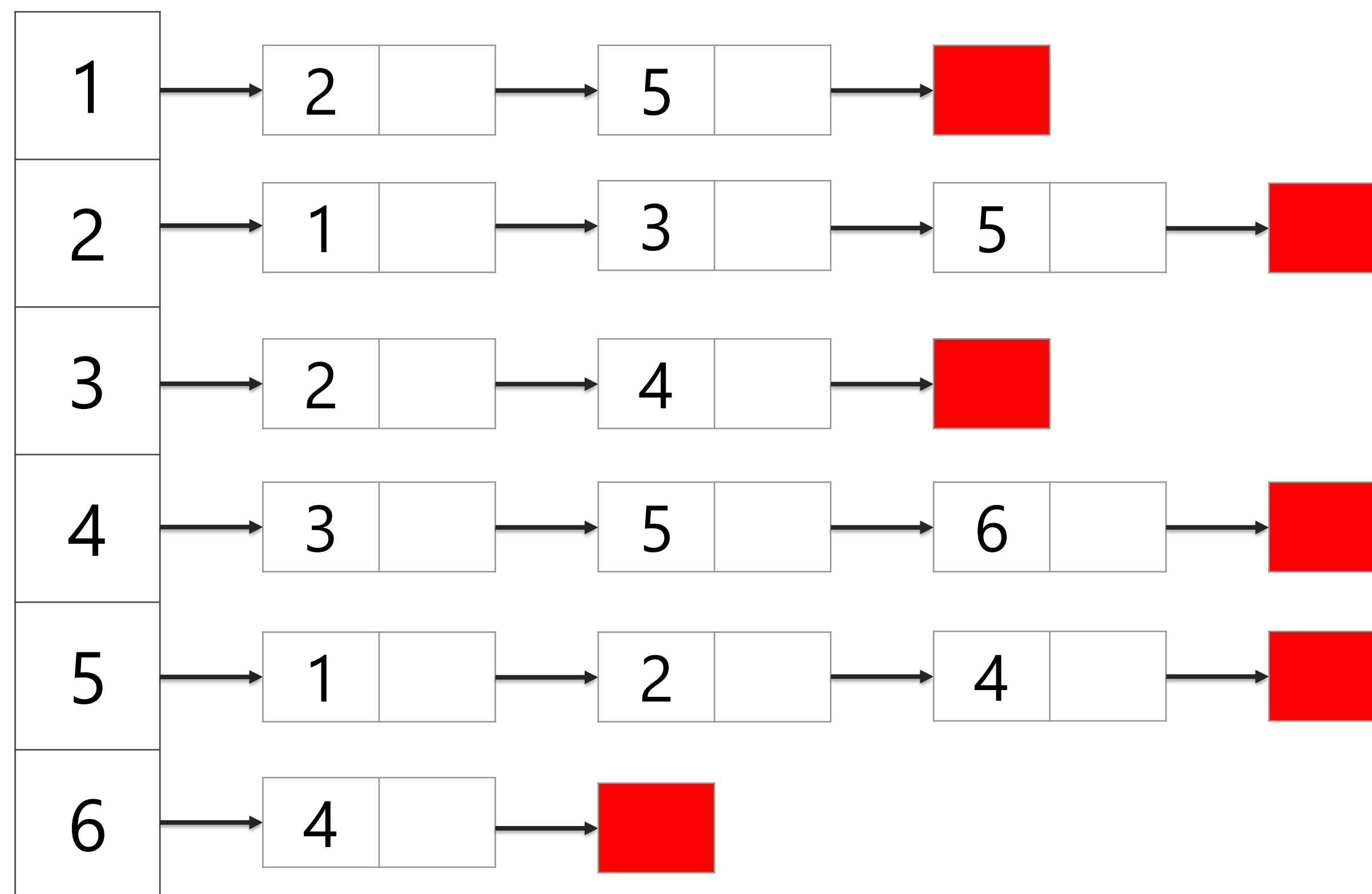
Sharing is learning

IV. Đồ thị



Sharing is learning

2. Biểu diễn đồ thị trên máy tính



IV. Đồ thị



Sharing is learning

2. Biểu diễn đồ thị trên máy tính

Danh sách cạnh (cung) – Incidence List:

- Một ma trận 2 cột (hoặc 3 cột hoặc nhiều cột) có nhiều dòng (số dòng phụ thuộc vào số cạnh).
- Dùng để lưu trữ các cặp đỉnh nối với nhau (lưu trữ đỉnh đầu và cuối của cạnh).
- Cột thứ 3 hoặc các cột khác dùng để lưu trọng số hoặc thông tin khác (nếu có).



Sharing is learning

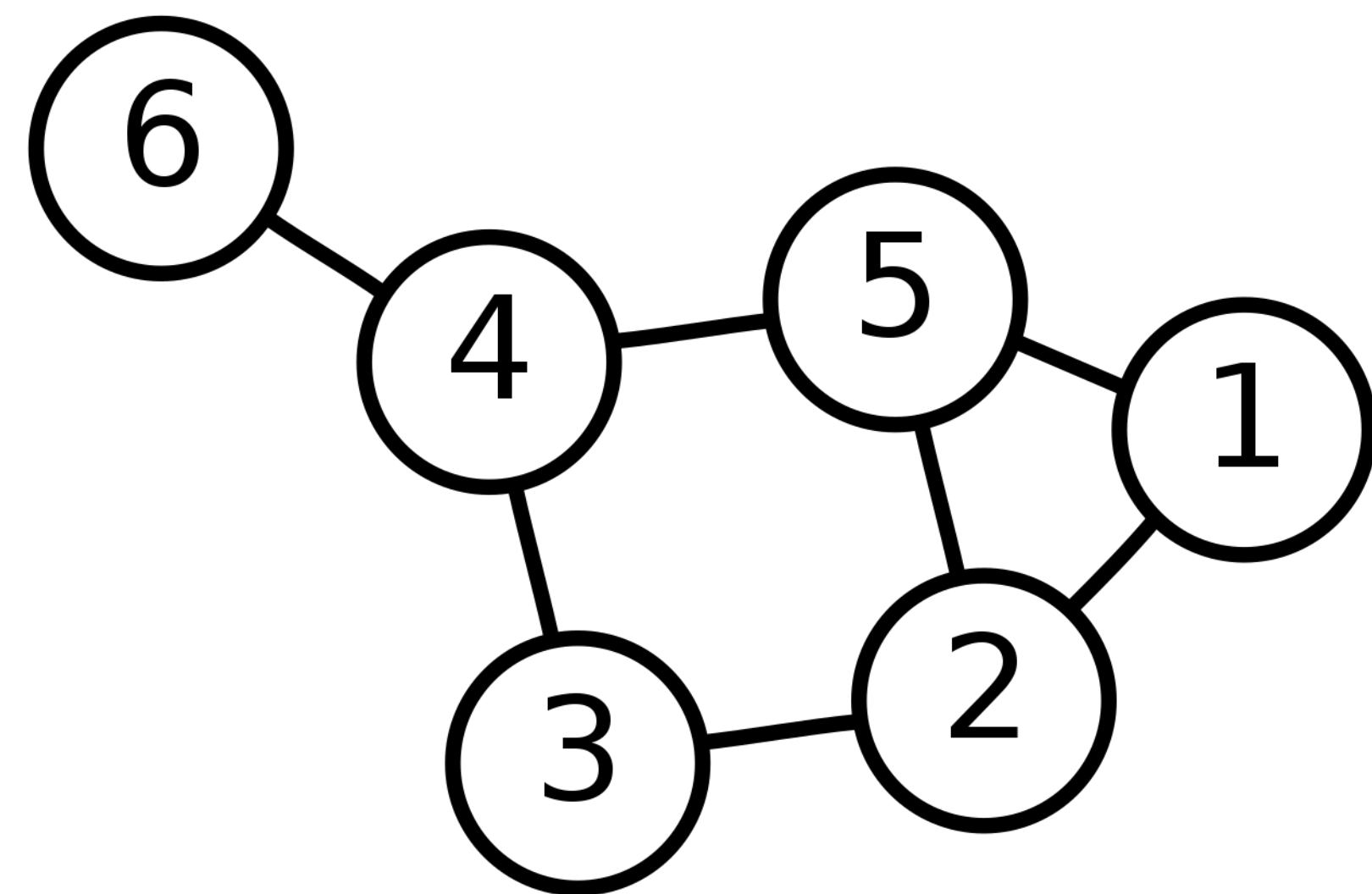
IV. Đồ thị



Sharing is learning

2. Biểu diễn đồ thị trên máy tính

| Đầu | Cuối |
|-----|------|
| 1 | 2 |
| 1 | 5 |
| 2 | 3 |
| 2 | 5 |
| 3 | 4 |
| 4 | 5 |
| 4 | 6 |



IV. Đồ thị

2. Biểu diễn đồ thị trên máy tính

Ma trận cạnh (cung) liên thuộc – Incidence Matrix:

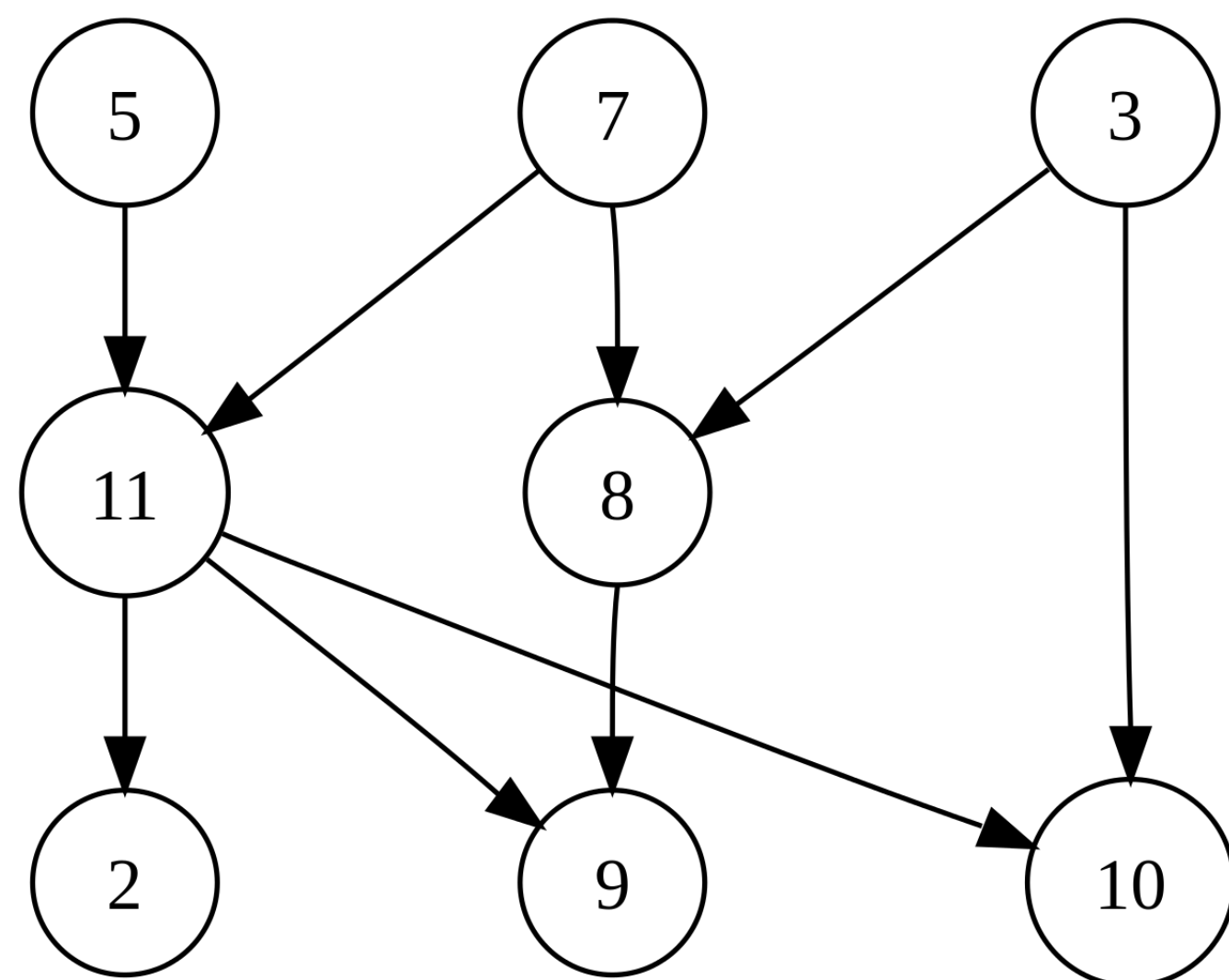
- Một ma trận gồm n dòng và m cột với:
 - ✓ n là số đỉnh.
 - ✓ m là số cạnh.
- Áp dụng cho đồ thị có hướng.
- Dùng để lưu trữ các trạng thái của 1 đỉnh so với 1 cạnh:
 - ✓ 1: Nếu đỉnh đó là đỉnh đầu của cạnh e .
 - ✓ -1: Nếu đỉnh đó là đỉnh cuối của cạnh e .
 - ✓ 0: Nếu đỉnh không thuộc cạnh đang xét.

IV. Đồ thị



Sharing is learning

2. Biểu diễn đồ thị trên máy tính



| | (3;8) | (3;10) | (5;11) | (7;8) | (7;11) | (8;9) | (11;2) | (11;9) | (11;10) |
|----|-------|--------|--------|-------|--------|-------|--------|--------|---------|
| 2 | | | | | | | -1 | | |
| 3 | 1 | 1 | | | | | | | |
| 5 | | | 1 | | | | | | |
| 7 | | | | 1 | 1 | | | | |
| 8 | -1 | | | -1 | | 1 | | | |
| 9 | | | | | | -1 | | -1 | |
| 10 | | -1 | | | | | | | -1 |
| 11 | | | -1 | | -1 | | 1 | 1 | 1 |

IV. Đồ thị



Sharing is learning

2. Biểu diễn đồ thị trên máy tính

| | |
|--|--|
| <i>Ma trận kề (ma trận liên kết) – Adjacency Matrix</i> | Biểu thị trạng thái, quan hệ giữa đỉnh với đỉnh (số cạnh nối/đường giữa các đỉnh). |
| <i>Danh sách kề - Adjacency List</i> | Biểu thị các đỉnh kề của 1 đỉnh. |
| <i>Danh sách cạnh (cung) – Incidence List</i> | Biểu thị các cạnh có trong đồ thị (đồ thị có hướng). |
| <i>Ma trận cạnh (cung) liên thuộc – Incidence Matrix</i> | Biểu thị trạng thái, quan hệ của đỉnh đối với cạnh. |

IV. Đồ thị



Sharing is learning

3. Duyệt đồ thị theo chiều sâu và chiều rộng

| Breadth-First Search | Depth-First Search |
|---|--|
| <ul style="list-style-type: none">Lần lượt lưu lại những đỉnh kề của đỉnh đang duyệt vào từ phía cuối tập đợi. Đánh dấu các đỉnh kề là đã duyệt.Lấy đỉnh từ phía đầu tập đợi để duyệt tiếp và tiếp tục cho đến khi tập đợi rỗng.Tập đợi có dạng là Queue. | <ul style="list-style-type: none">Nếu tồn tại đỉnh kề chưa được duyệt thì chọn 1 đỉnh kề chưa được duyệt của đỉnh đang xét để duyệt tiếp và lưu đỉnh đang xét vào từ phía đầu tập đợi và đánh dấu đỉnh đang xét là đã duyệt.Lấy đỉnh từ phía đầu tập đợi để duyệt tiếp và tiếp tục cho đến khi tập đợi rỗng.Tập đợi có dạng là Stack. |

V. Đồ thị



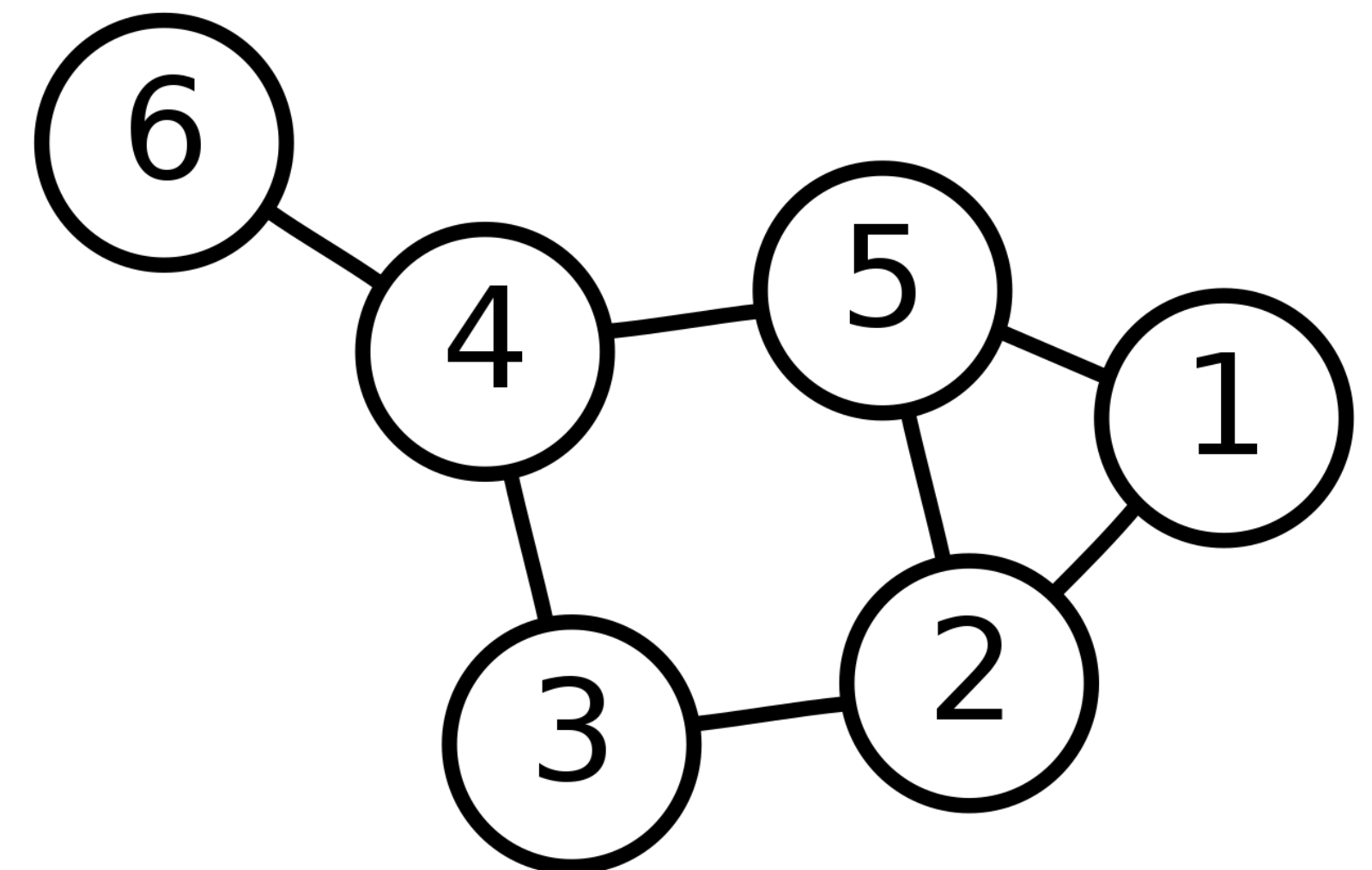
Sharing is learning

3. Duyệt đồ thị theo chiều sâu VÀ chiều rộng

Breath-First Search:

Select Vertex: 1

| | | | | | | |
|---------|---|---|---|---|---|---|
| Visited | 1 | 2 | 5 | 3 | 4 | 6 |
| Queue | 1 | | | | | |
| Queue | 2 | 5 | | | | |
| Queue | 5 | 3 | | | | |
| Queue | 3 | 4 | | | | |
| Queue | 4 | | | | | |
| Queue | 6 | | | | | |
| Queue | | | | | | |



| | | | | | | |
|--------|---|---|---|---|---|---|
| Output | 1 | 2 | 5 | 3 | 4 | 6 |
|--------|---|---|---|---|---|---|

IV. Đồ thị



Sharing is learning

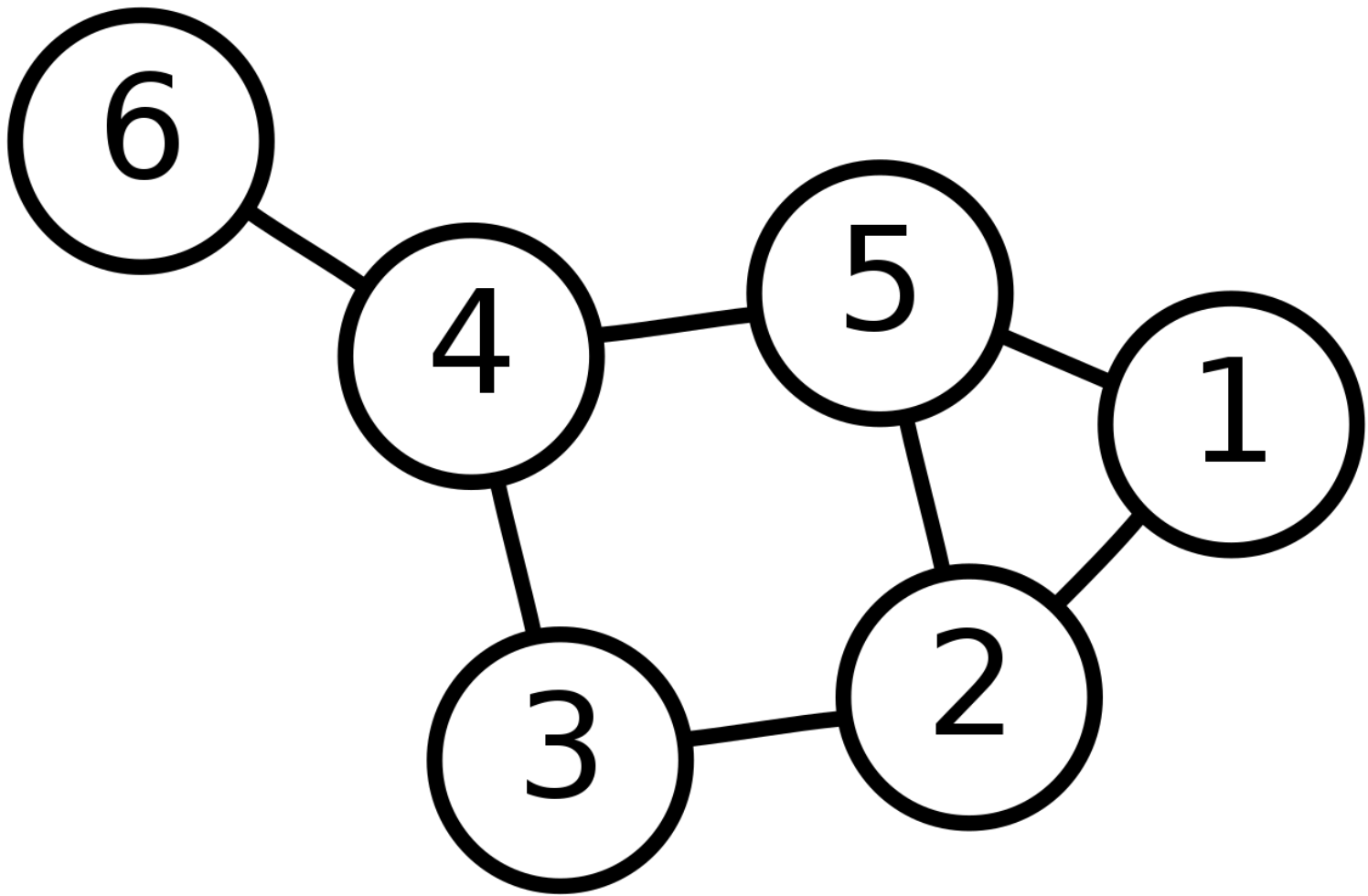
3. Duyệt đồ thị theo chiều sâu và chiều rộng

Depth-First Search:

Select Vertex: 1

| | | | | | | |
|---------|---|---|---|---|---|---|
| Visited | 1 | 2 | 3 | 4 | 5 | 6 |
| Output | 1 | 2 | 3 | 4 | 5 | 6 |

| | | | | | | | | | |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Select | 2 | 3 | 4 | 5 | 6 | x | x | x | x |
| | | | | | | | | | |
| | | | | 4 | 4 | | | | |
| | | | 3 | 3 | 3 | 3 | | | |
| | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| Stack | Stack | Stack | Stack | Stack | Stack | Stack | Stack | Stack | Stack |



Nội dung Training

I. Các thuật toán sắp xếp

II. Cây nhị phân

III. Bảng băm

IV. Đồ thị

V. Giải đề



Sharing is learning



Sharing is learning

VI. Giải đề



Sharing is learning

1. Câu 1

- a. Hãy trình bày các bước giải thuật sắp xếp **chèn trực tiếp** để sắp xếp một dãy số nguyên **giảm dần** (không viết hàm) và cho biết độ phức tạp của giải thuật.
- b. Cho dãy số nguyên A sau: 20; 40; 65; 0; 1. Hãy chạy từng bước theo giải thuật ở câu 1.a để sắp xếp dãy số A giảm dần.

VI. Giải đề



Sharing is learning

1.Câu 1a

- Bước 1: $i = 1$;
- Bước 2: $x = a[i]$; Tìm vị trí pos thích hợp trong đoạn $a[1]$ đến $a[i-1]$ để chèn $a[i]$ vào
- Bước 3: Dời chỗ các phần tử từ $a[pos]$ đến $a[i-1]$ sang phải 1 vị trí để dành chỗ cho $a[i]$
- Bước 4: $a[pos] = x$; //có đoạn $a[1]..a[i]$ đã được sắp
- Bước 5: $i = i + 1$;
Nếu $i < n$: Lặp lại Bước 2
Ngược lại : Dừng

Độ phức tạp:

| Tốt nhất | Trung bình | Xấu nhất |
|----------|------------|----------|
| $O(n)$ | $O(n^2)$ | $O(n^2)$ |

VI. Giải đề



Sharing is learning

1.Câu 1b

$A = 20 \ 40 \ 65 \ 0 \ 1$

- $i = 1$
 $x = A[1] = 40$
 $pos = 0$

$A = 40 \ 20 \ 65 \ 0 \ 1$

- $i = 2$
 $x = A[2] = 65$
 $pos = 0$

$A = 65 \ 40 \ 20 \ 0 \ 1$

- $i = 3$
 $x = 0$
 $pos = 3$

$A = 65 \ 40 \ 20 \ 0 \ 1$

- $i = 4$
 $x = 1$
 $pos = 3$

$A = 65 \ 40 \ 20 \ 1 \ 0$

VI. Giải đề



Sharing is learning

2. Câu 2

Cho dãy số nguyên B: 46; 35; 14; 28; 57; 75; 60; 82; 79; 96; 20; 99; 97

Hãy thực hiện các yêu cầu:

- Vẽ **cây nhị phân tìm kiếm** bằng cách thêm lần lượt từng số nguyên vào cây theo thứ tự từ trái qua phải của dãy số B.
- Vẽ cây khi hủy lần lượt nút 35; 14; 79; 46 trên cây, mỗi lần hủy 1 nút vẽ lại cây.
- Cho biết kết quả duyệt cây theo RNL, NRL.
- Viết hàm đếm số nút trên cây có giá trị lớn hơn 30.

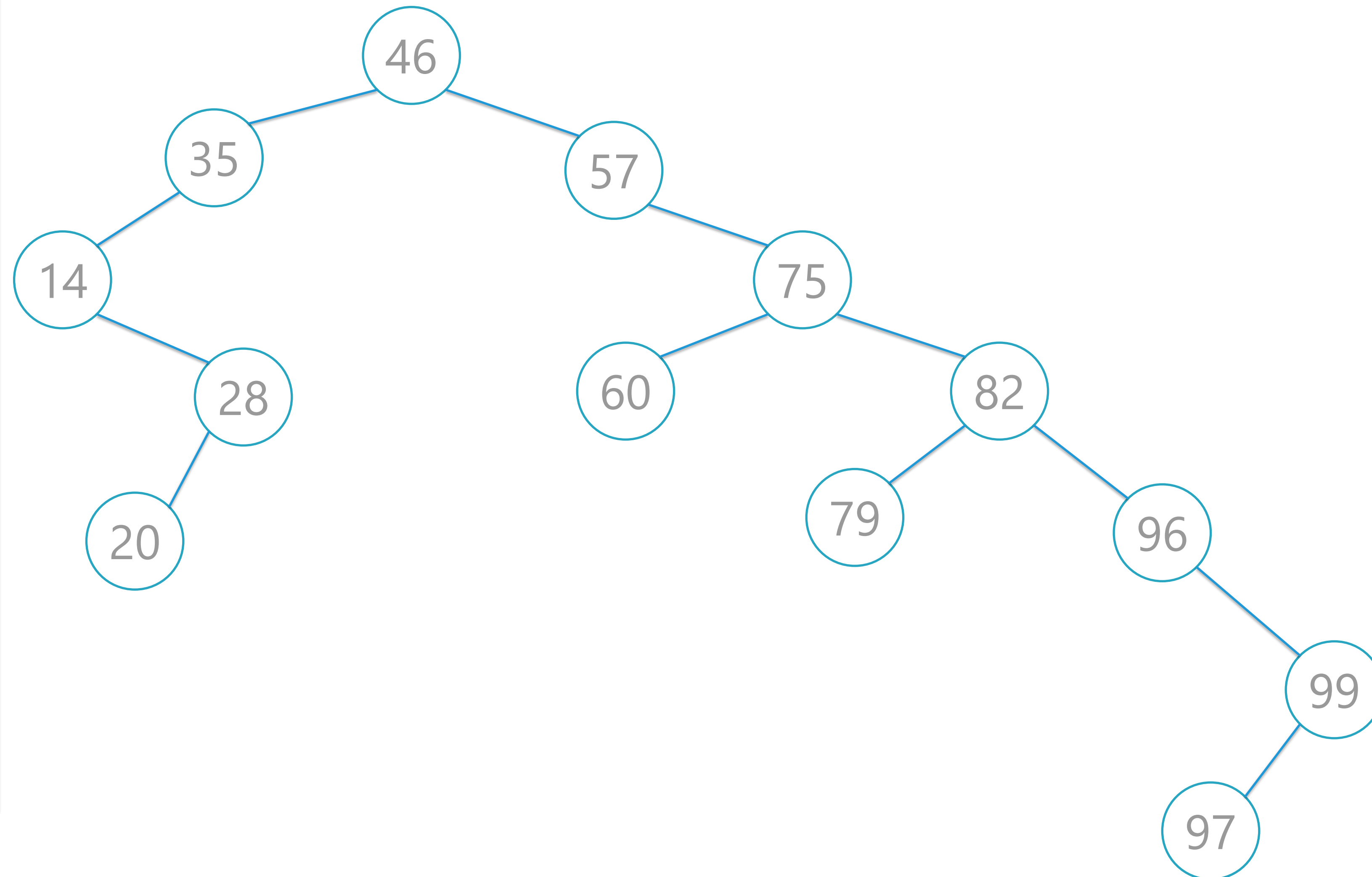
VI. Giải đề



Sharing is learning

2. Câu 2a

| | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 46 | 35 | 14 | 28 | 57 | 75 | 60 | 82 | 79 | 96 | 20 | 99 | 97 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|



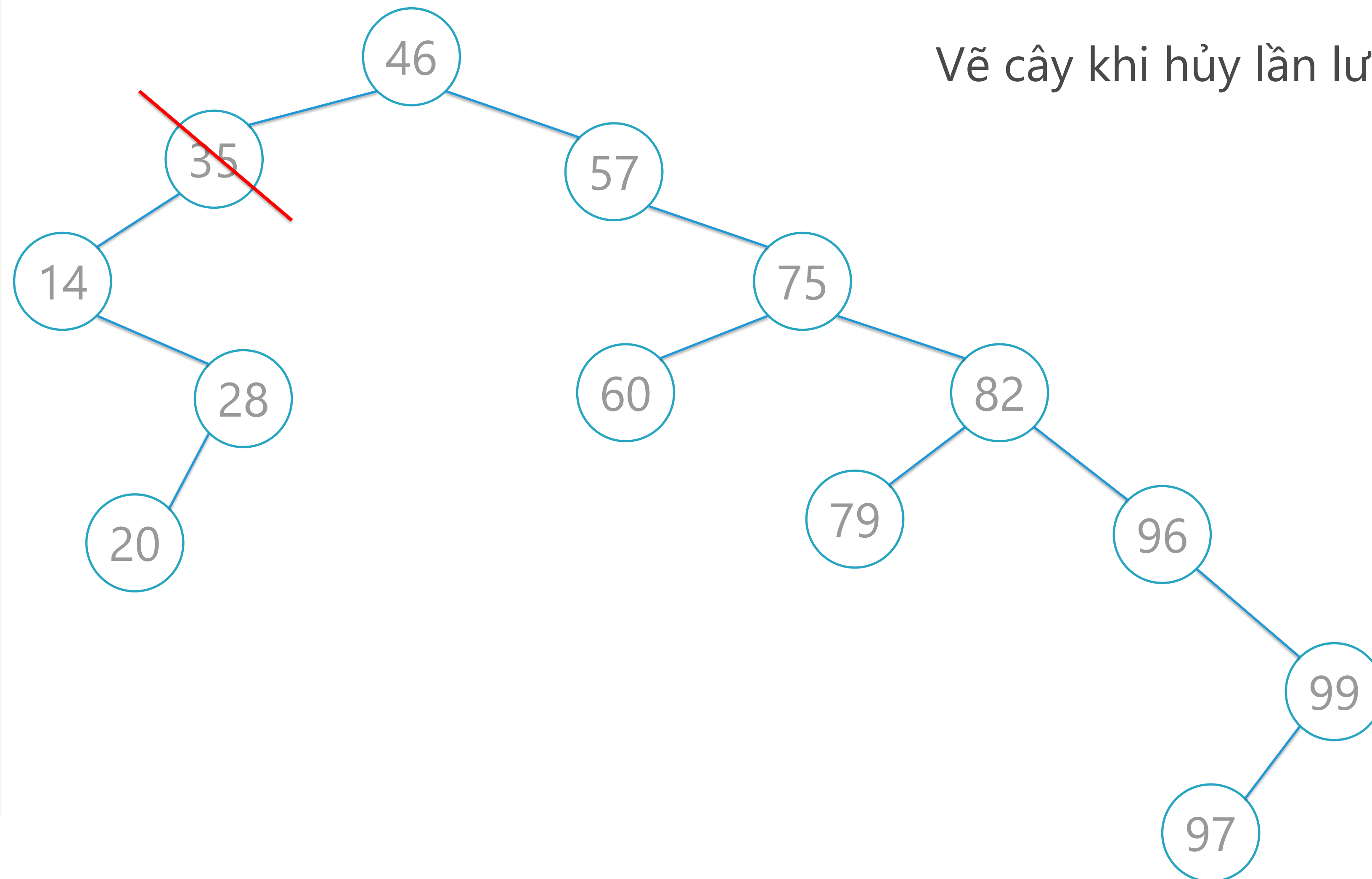
VI. Giải đề



Sharing is learning

2. Câu 2b

Vẽ cây khi hủy lần lượt nút 35



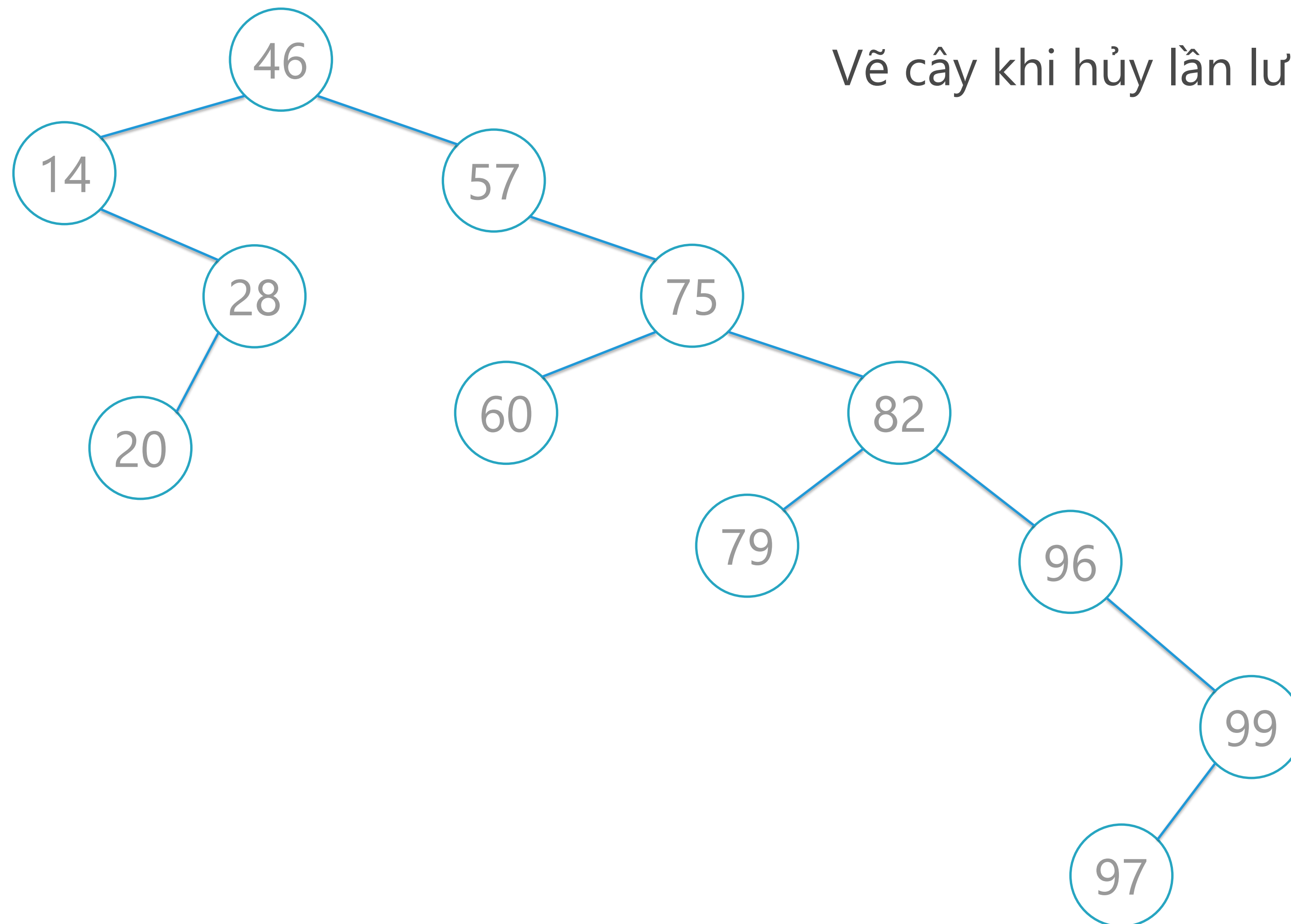
VI. Giải đề



Sharing is learning

2. Câu 2b

Vẽ cây khi hủy lần lượt nút 35



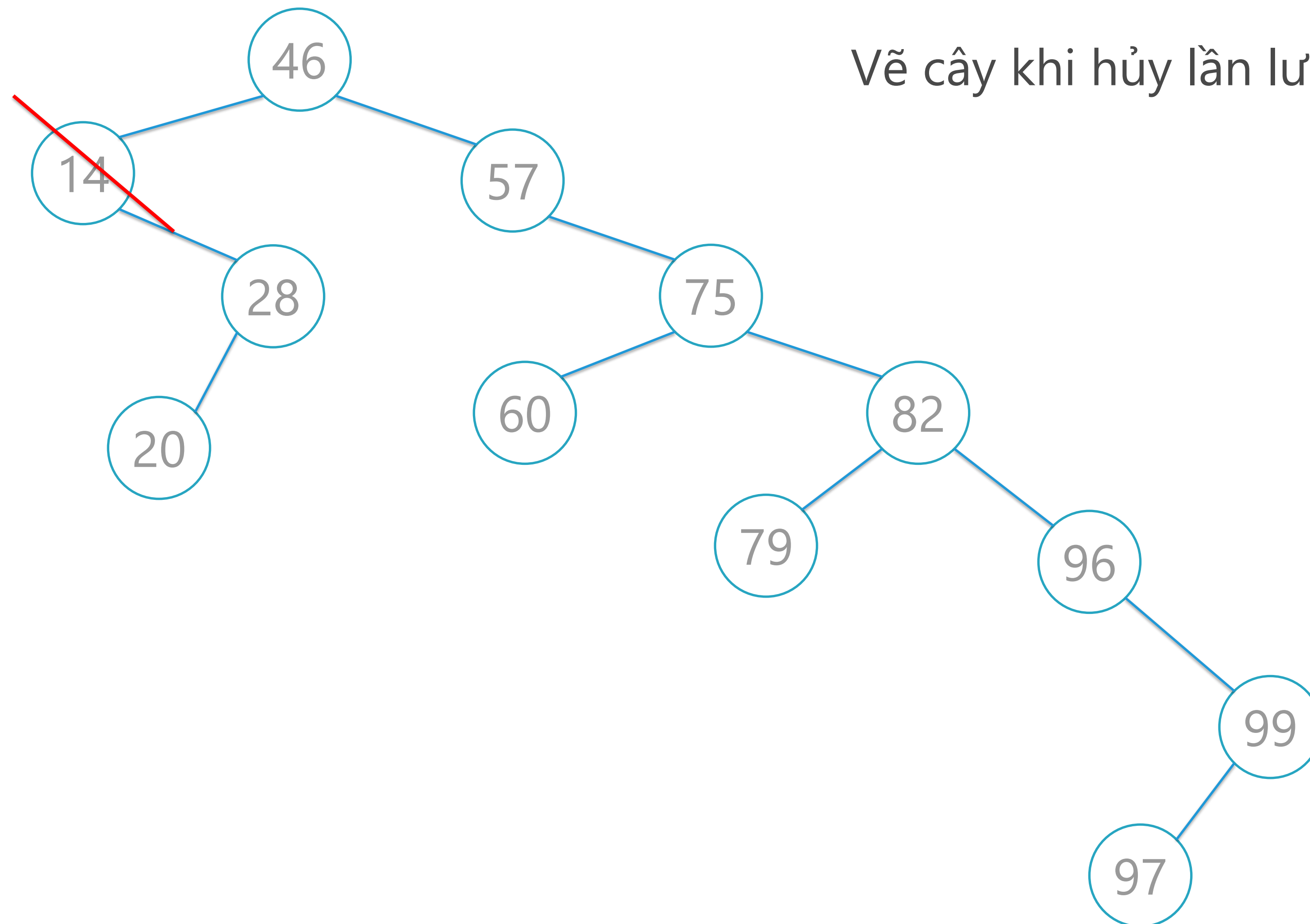
VI. Giải đề



Sharing is learning

2. Câu 2b

Vẽ cây khi hủy lần lượt nút 14

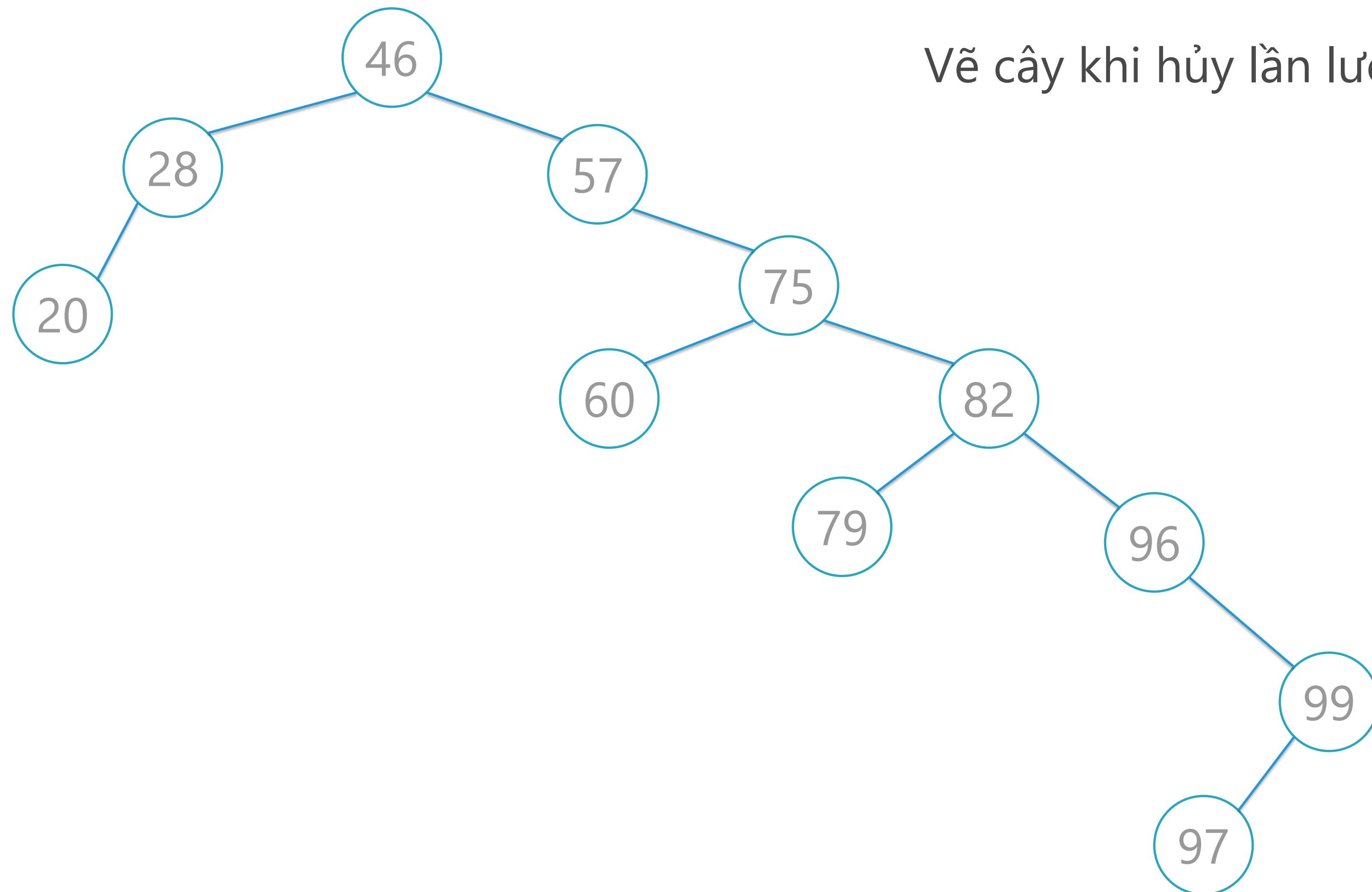


VI. Giải đề



Sharing is learning

2. Câu 2b



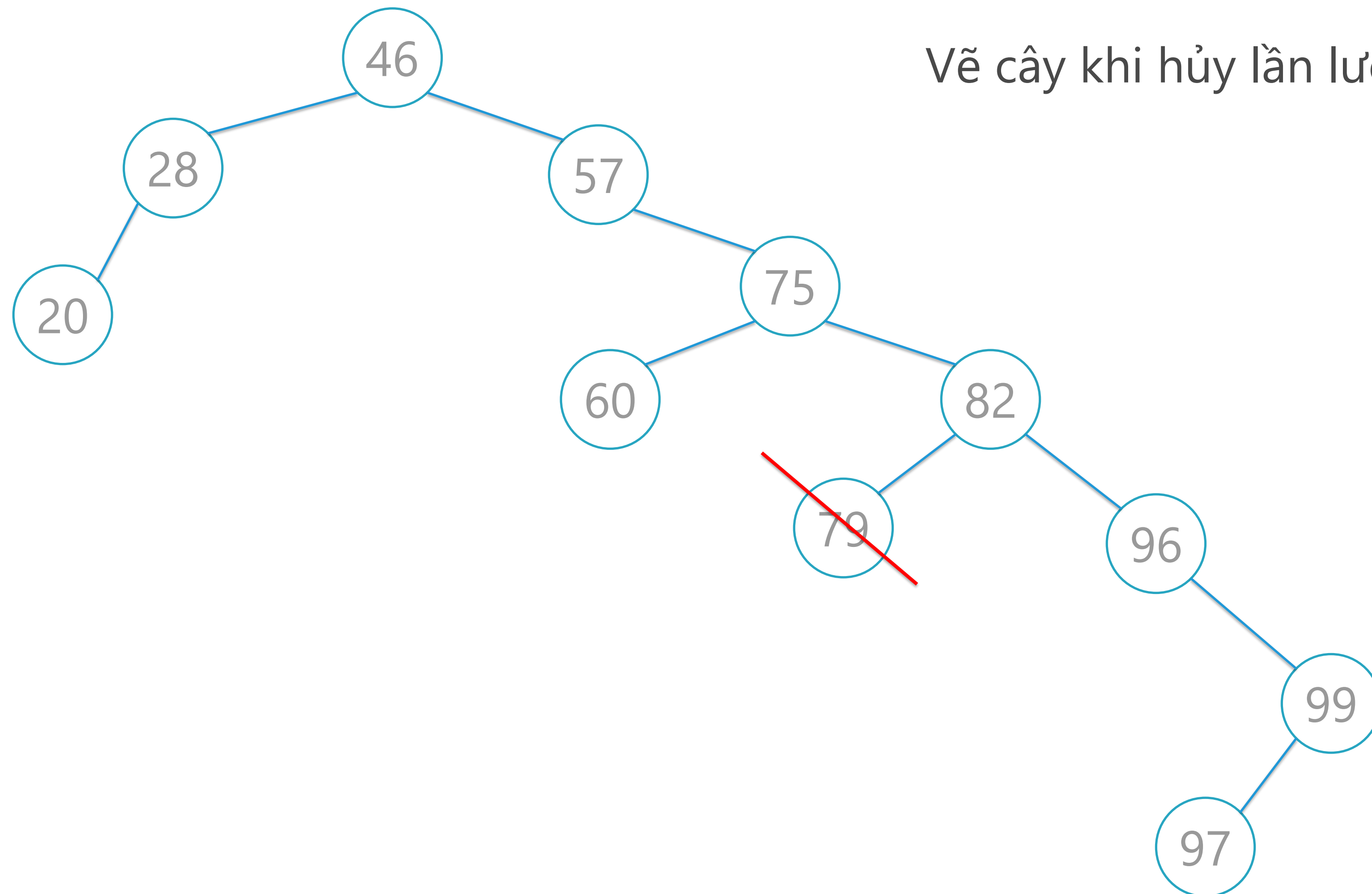
VI. Giải đề



Sharing is learning

2. Câu 2b

Vẽ cây khi hủy lần lượt nút 79

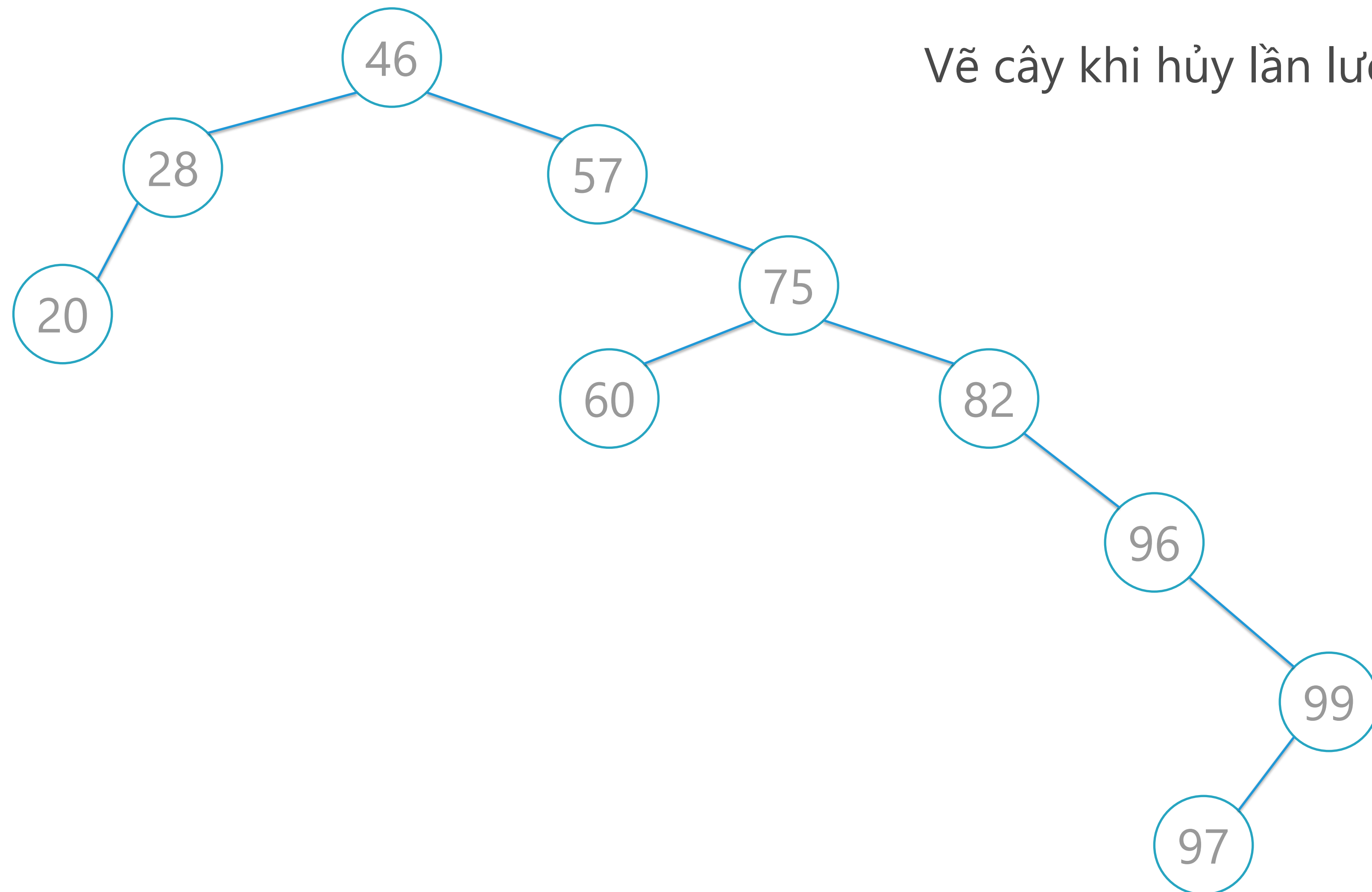


VI. Giải đề



Sharing is learning

2. Câu 2b

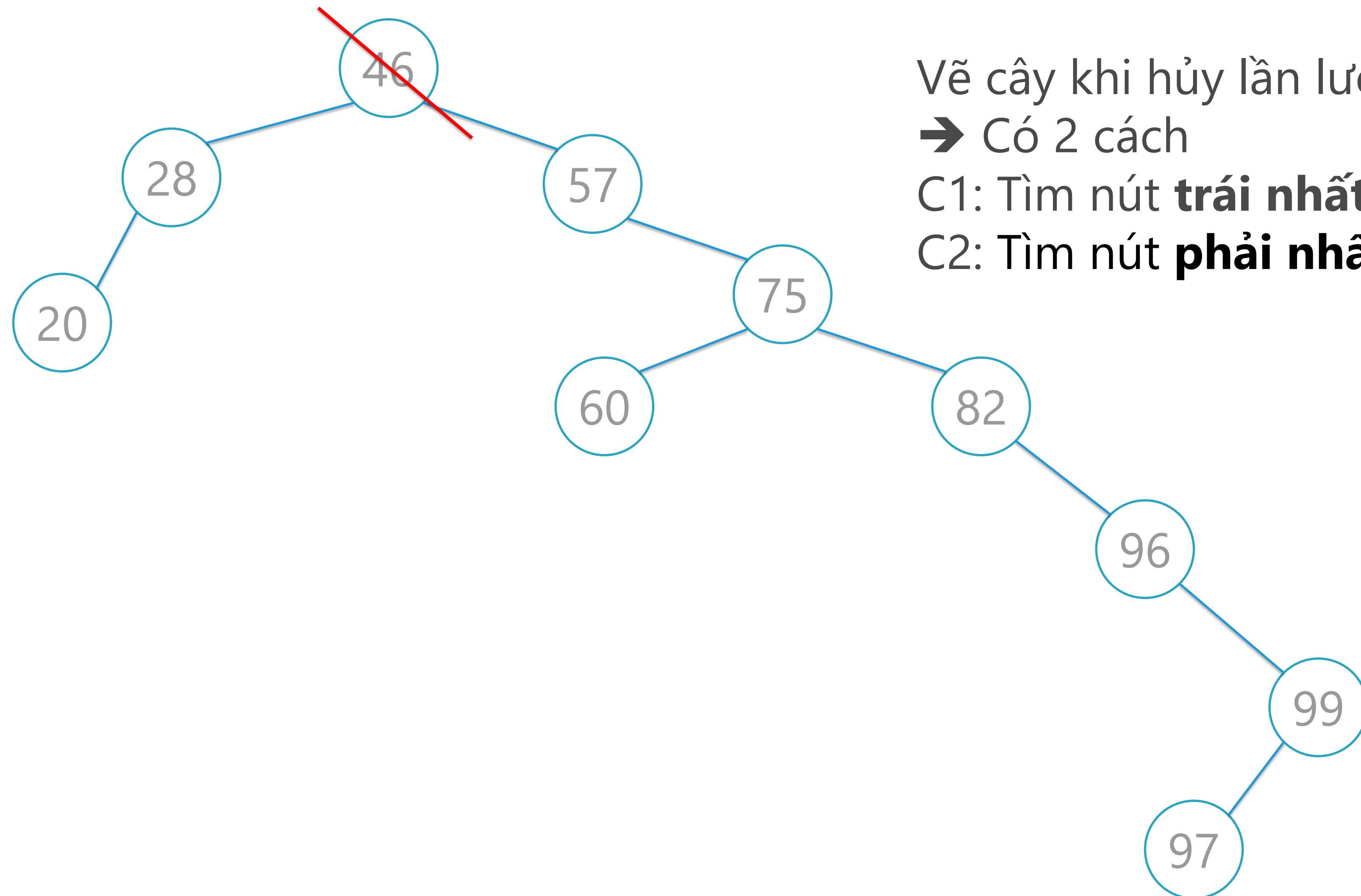


VI. Giải đề



Sharing is learning

2. Câu 2b



Vẽ cây khi hủy lần lượt nút 46

→ Có 2 cách

C1: Tìm nút **trái nhất** của **cây con bên phải**.

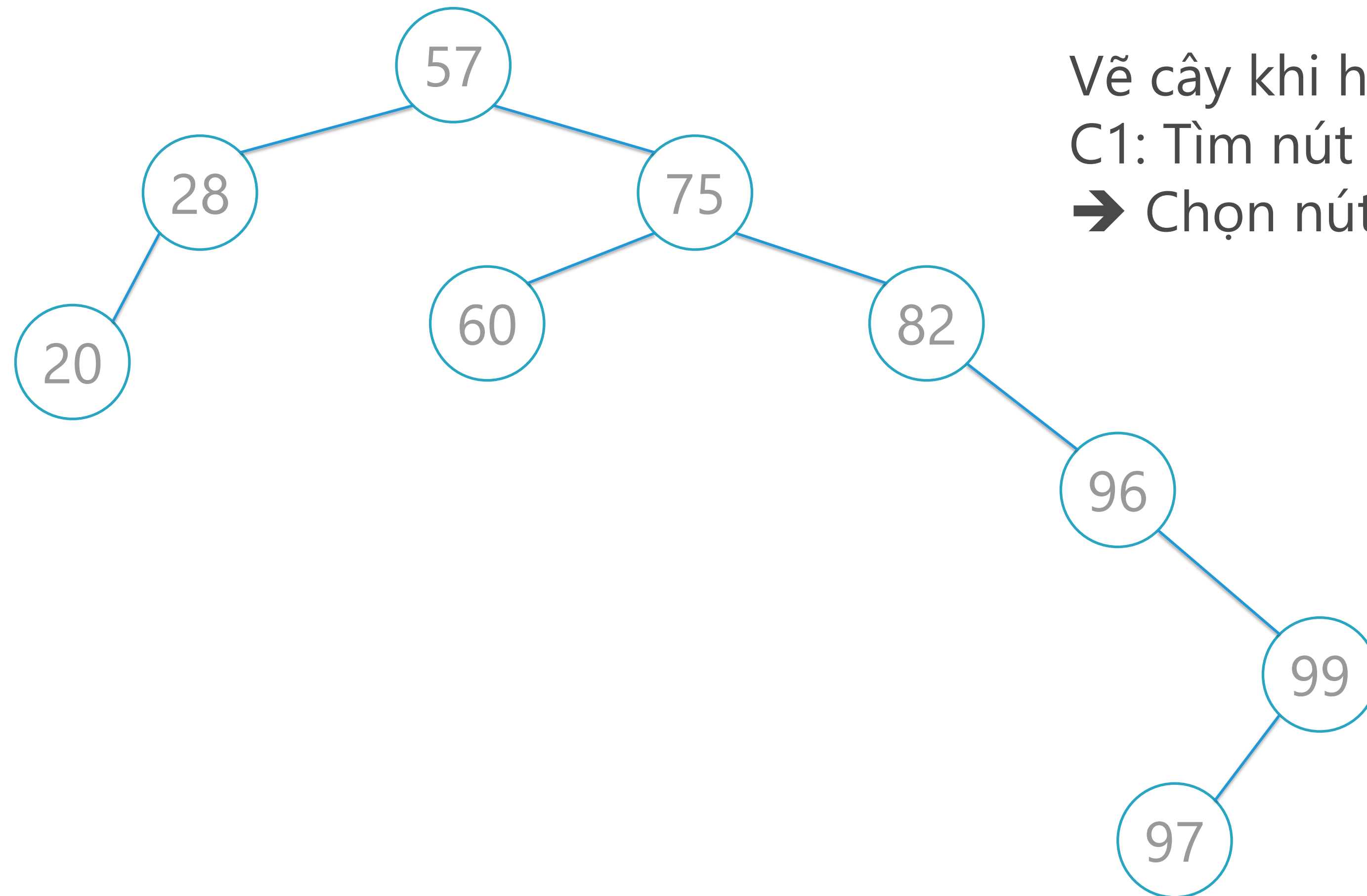
C2: Tìm nút **phải nhất** của **cây con bên trái**.

VI. Giải đề



Sharing is learning

2. Câu 2b



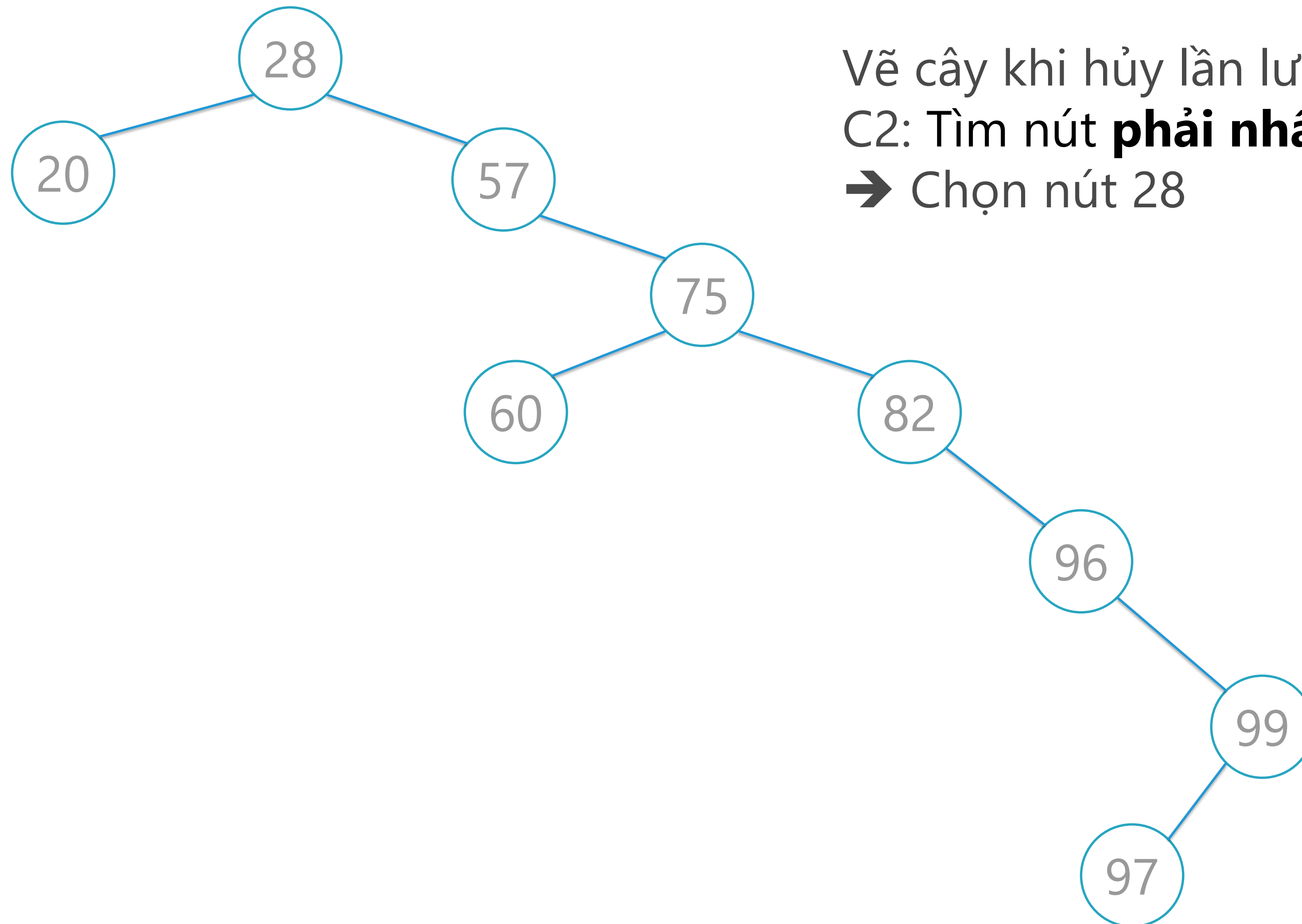
Vẽ cây khi hủy lần lượt nút 46 (**cách 1**)
C1: Tìm nút **trái nhất** của **cây con bên phải**.
→ Chọn nút 57

VI. Giải đề



Sharing is learning

2. Câu 2b



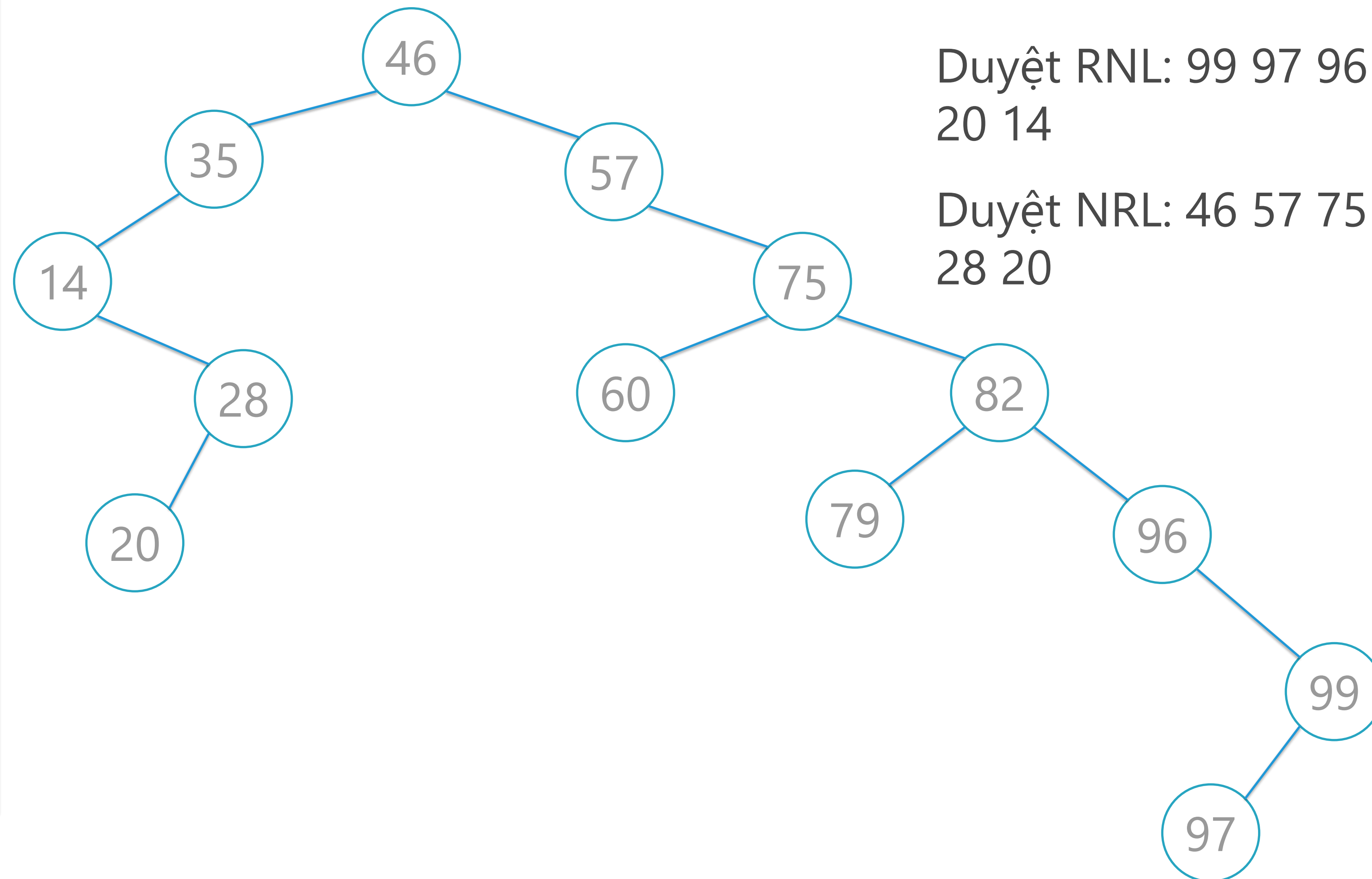
Vẽ cây khi hủy lần lượt nút 46 (**cách 2**)
C2: Tìm nút **phải nhất** của **cây con bên trái**.
→ Chọn nút 28

VI. Giải đề



Sharing is learning

2. Câu 2c



Duyệt RNL: 99 97 96 82 79 75 60 57 46 35 28
20 14

Duyệt NRL: 46 57 75 82 96 99 97 79 60 35 14
28 20

VI. Giải đề



Sharing is learning

2. Câu 2d

```
struct TNode
{
    int key;
    TNode* Left;
    TNode* Right;
};

void Dem_so_lon_hon_30(TNode *T, int& dem)
{
    if (T != NULL)
    {
        if (T->key > 30)
            dem++;
        Dem_so_lon_hon_30(T->Right, dem);
        Dem_so_lon_hon_30(T->Left, dem);
    }
}
```

VI. Giải đề



Sharing is learning

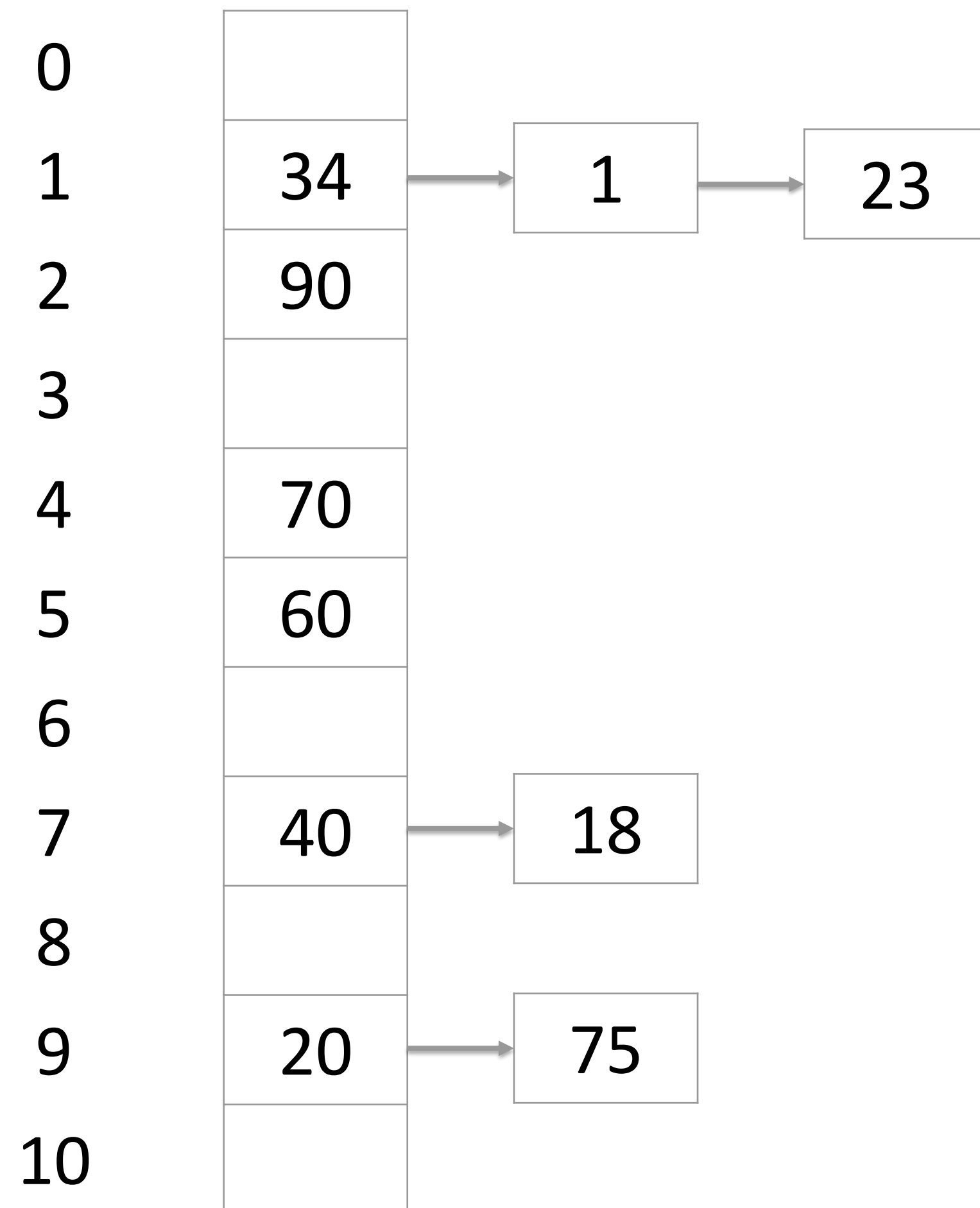
3. Câu 3

Cho dãy số nguyên C sau: 40; 34; 18; 20; 1; 70; 60; 23; 75; 90 và bảng băm gồm 11 ô trống. Hàm băm $h(\text{key}) = \text{key} \bmod 11$ ($\text{key} \% 11$)

Hãy vẽ hình bảng băm khi thêm lần lượt vào bảng băm các số nguyên trong dãy C theo thứ tự từ trái qua phải. Trong trường hợp đụng độ, giải quyết bằng phương **pháp nối kết trực tiếp** (Direct/Separate chaining).

VI. Giải đề

| | | | | | | | | | |
|----|----|----|----|---|----|----|----|----|----|
| 40 | 34 | 18 | 20 | 1 | 70 | 60 | 23 | 75 | 90 |
|----|----|----|----|---|----|----|----|----|----|



- $40 \% 11 = 7$
→ Vào vị trí số 7
- $34 \% 11 = 1$
→ Vào vị trí số 1
- $18 \% 11 = 7$
→ Xảy ra đụng độ
→ Chèn vào danh sách liên kết ở vị trí số 7
- $20 \% 11 = 9$
→ Vào vị trí số 9
- $1 \% 11 = 1$
→ Xảy ra đụng độ
→ Chèn vào danh sách liên kết ở vị trí số 1

- $70 \% 11 = 4$
→ Vào vị trí số 4
- $60 \% 11 = 5$
→ Vào vị trí số 5
- $23 \% 11 = 1$
→ Xảy ra đụng độ
→ Chèn vào danh sách liên kết ở vị trí số 1
- $75 \% 11 = 9$
→ Xảy ra đụng độ
→ Chèn vào danh sách liên kết ở vị trí số 9
- $90 \% 11 = 2$
→ Vào vị trí số 2

BAN HỌC TẬP KHOA CÔNG NGHỆ PHẦN MỀM

CHUỖI TRAINING CUỐI HỌC KÌ 2 NĂM HỌC 2021 - 2022



Sharing is learning

HẾT

**CẢM ƠN CÁC BẠN ĐÃ THEO DÕI.
CHÚC CÁC BẠN CÓ KẾT QUẢ THI THẬT TỐT!**



Ban học tập

Khoa Công Nghệ Phần Mềm
Trường ĐH Công Nghệ Thông Tin
ĐHQG Hồ Chí Minh



Email / Group

bht.cnpm.uit@gmail.com
fb.com/groups/bht.cnpm.uit