

Bài 6.4: Cập nhật một node trong heap

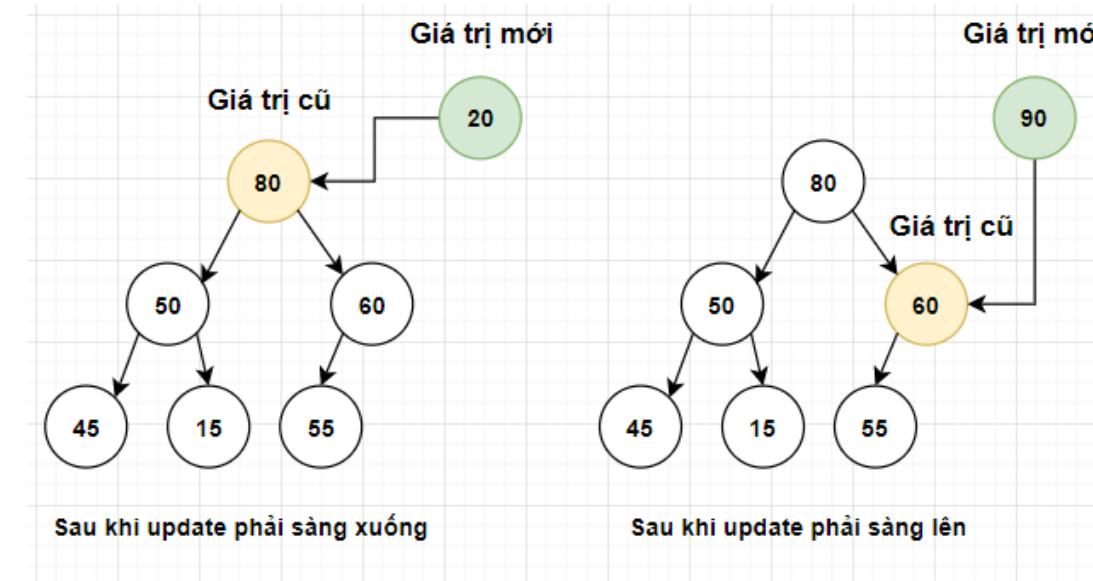
- ✓ Các bước thực hiện
- ✓ Mã giả và triển khai
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

Các bước thực hiện

- Yêu cầu: cập nhật giá trị của một node trong max heap.
- Các bước thực hiện:
 - B1: tìm vị trí node cần sửa gọi là index.
 - B2: nếu tìm thấy, gán giá trị mới cho node tại vị trí index.
 - B3: tìm vị trí node cha của node vừa cập nhật.
 - B4: nếu node cha < node tại vị trí index, sàng lên.
 - B5: ngược lại, sàng xuống để tái cân bằng lại heap.

Các bước thực hiện

➤ Hình ảnh minh họa:



Cập nhật node trong heap

➤ Sau đây là mã giả thao tác cập nhật node trong heap:

```
// thao tác cập nhật một node trong heap
bool update(oldNode, newNode) { // thay thế oldNode bởi newNode
    index = findNode(oldNode); // tìm vị trí của node cũ
    if (index >= 0) { // nếu tìm thấy
        data[index] = newNode; // cập nhật node cũ bằng node mới
        parentIndex = (index - 1) / 2; // tìm vị trí node cha
        // cân bằng lại heap sau khi update
        if (data[parentIndex] < data[index]) { // nếu node cha nhỏ hơn
            siftUp(index); // sàng lên
        } else { // ngược lại
            siftDown(index); // sàng xuống
        }
    }
}
```

Cập nhật node trong heap

➤ Sau đây là mã thật thao tác cập nhật node trong heap:

```
/*
 * Phương thức cập nhật một node trong heap
 *
 * @param oldNode node cũ cần cập nhật
 * @param newNode giá trị node mới dùng để thay thế cho node cũ
 * @return true nếu cập nhật thành công và false trong trường hợp ngược lại
 */
public boolean update(E oldNode, E newNode) {
    var index : int = findNode(oldNode);
    if (index >= 0) { // nếu tìm thấy giá trị cần cập nhật
        data[index] = newNode;
        var parentIndex = (index - 1) / 2;
        if (data[parentIndex].compareTo(data[index]) < 0) {
            siftUp(index);
        } else {
            siftDown(index);
        }
        return true; // cập nhật thành công
    }
    return false; // cập nhật thất bại
}
```

Nội dung tiếp theo

Tạo hàng đợi ưu tiên dùng heap