

## Bài 7.4: Cập nhật một node trong heap

---

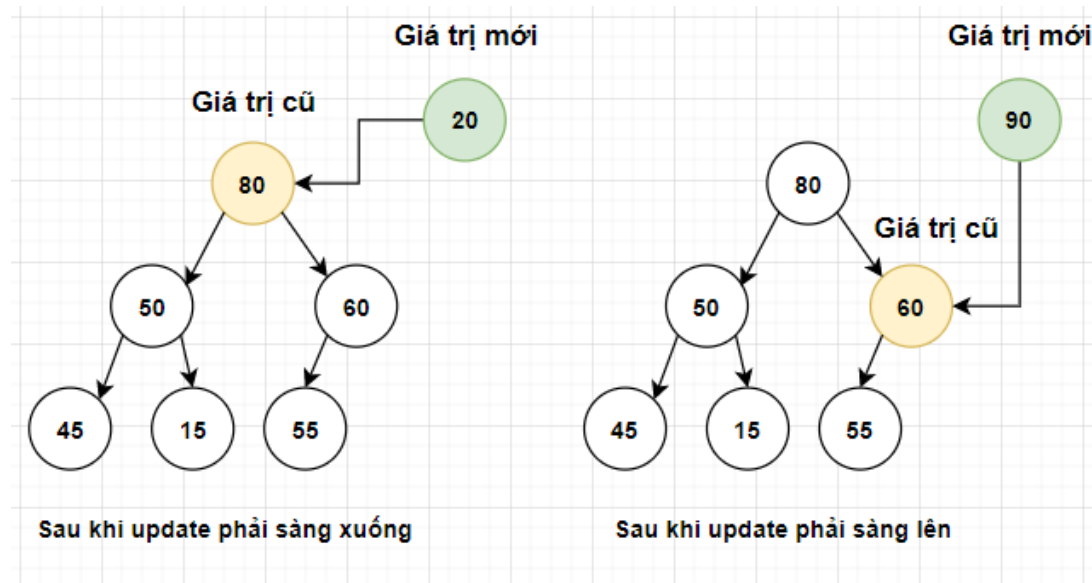
- ✓ Các bước thực hiện
- ✓ Mã giả và triển khai
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

# 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 node trong heap
// oldNode: node cũ
// newNode: node mới
function update(oldNode, newNode):
    index = findNodeIndex(oldNode) //tìm vị trí node cũ
    if(index >= 0): // nếu tìm thấy node cũ
        data[index] = newNode // cập nhật thành node mới
        parentIndex = (index - 1) / 2 // tìm vị trí node cha
        // cân bằng lại heap sau khi update
        if(data[parentIndex] < data[index]): // node mới > cha
            siftUp(index) // sàng lên
        else: // node mới nhỏ hơn hoặc bằng node cũ
            siftDown(index) // sàng xuống
```

# Cập nhật node trong heap

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

```
// cập nhật node trong heap
bool update(T oldNode, T newNode) {
    int index = findNodeIndex(oldNode);
    if (index >= 0) {
        data[index] = newNode;
        int parentIndex = (index - 1) / 2;
        if (data[parentIndex] < data[index]) {
            siftUp(index);
        }
        else {
            siftDown(index);
        }
        return true; // cập nhật thành công
    }
    return false; // không tìm thấy, cập nhật thất bại
}
```

# Nội dung tiếp theo

Tạo hàng đợi ưu tiên dùng heap