

# Thuật toán cây đỏ đen (Red-black tree)

Cây đỏ đen (Red-black tree) là một cấu trúc dữ liệu dạng cây nhị phân tương tự như cây nhị phân tìm kiếm (BST). Tuy nhiên, cây đỏ đen được thiết kế để giảm thiểu chi phí của các phép cập nhật, bao gồm cả chèn và xóa. Các nút trong cây đỏ đen được đánh dấu bằng màu đỏ hoặc màu đen, và phải tuân thủ một số quy tắc nhất định để đảm bảo rằng cây luôn cân bằng và có độ cao  $O(\log n)$ .

## Quy tắc

Các quy tắc sau đây phải được tuân thủ để đảm bảo rằng cây đỏ đen luôn cân bằng:

1. Mỗi nút là màu đỏ hoặc đen.
2. Nút gốc luôn luôn phải là màu đen.
3. Nếu một nút là màu đỏ, thì hai nút con của nó phải là màu đen.
4. Tất cả các đường đi từ nút gốc đến lá phải có số lượng nút đen bằng nhau.
5. Nếu một nút là lá, thì nó phải là màu đen.

## Các thao tác trên cây đỏ đen

Các thao tác trên cây đỏ đen bao gồm:

- Chèn (insert)
- Xóa (delete)
- Tìm kiếm (search)

Các thao tác này được triển khai bằng các hàm phù hợp. Dưới đây là đoạn mã Python mẫu cho các thao tác trên cây đỏ đen:

```
class Node:
    def __init__(self, val, color='RED'):
        self.val = val
        self.color = color
        self.left = None
        self.right = None
        self.parent = None
```

```
class RedBlackTree:
```

```
def __init__(self):
    self.root = None

def insert(self, val):
    new_node = Node(val, 'RED')
    self._insert(new_node)

def _insert(self, node):
    if self.root is None:
        self.root = node
        node.color = 'BLACK'
        return

    current = self.root
    parent = None

    while current is not None:
        parent = current
        if node.val < current.val:
            current = current.left
        else:
            current = current.right

    node.parent = parent

    if parent is None:
        self.root = node
        node.color = 'BLACK'
        return

    if parent.color == 'BLACK':
        return

    grandparent = parent.parent
    if parent == grandparent.left:
        uncle = grandparent.right
        if uncle is not None and uncle.color == 'RED':
            parent.color = 'BLACK'
            uncle.color = 'BLACK'
            grandparent.color = 'RED'
```

```
        self._insert(grandparent)
    else:
        if node == parent.right:
            self._rotate_left(parent)
            node, parent = parent, node
        self._rotate_right(grandparent)
        parent.color = 'BLACK'
        grandparent.color = 'RED'
    else:
```

---

Revision #1

Created 27 June 2023 10:57:21 by Laptrinh.vn

Updated 27 June 2023 10:59:15 by Laptrinh.vn