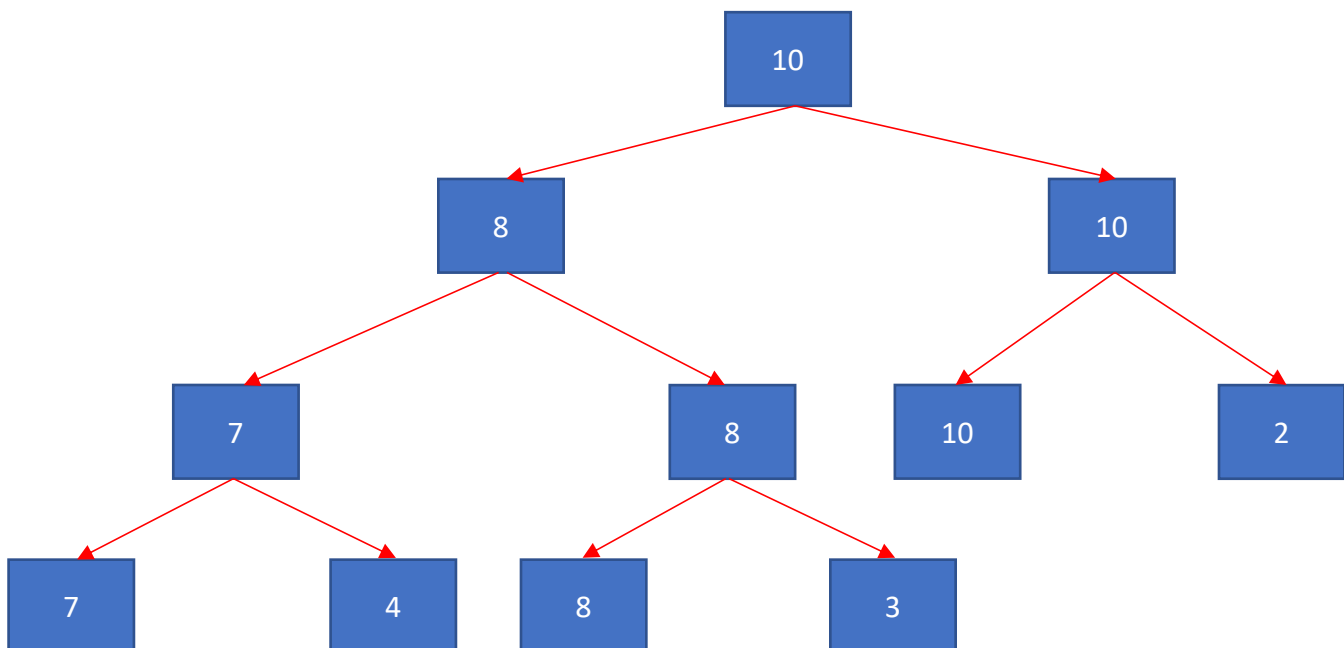


Segment tree

1. Segment tree là gì? Khi nào nên dùng Segment Tree? Ưu nhược điểm của Segment tree?
 - Segment tree là một cấu trúc dữ liệu giúp thực hiện các truy vấn trong một phạm vi của một mảng một cách hiệu quả, trong khi vẫn rất linh hoạt để cho phép sửa đổi mảng (với độ phức tạp $O(\log(n))$).
 - Cây phân đoạn được dùng khi ta cần tương tác với dữ liệu số. Nó được dùng để thực hiện tính tổng hoặc tìm giá trị lớn nhất, nhỏ nhất,...
 - Ưu điểm:
 - o Cây phân đoạn rút ngắn thời gian truy vấn cũng như sửa đổi rất linh hoạt.
 - Nhược điểm:
 - o Thao tác xây dựng khá cầu kì
 - o Segment tree đã bị hạn chế về phạm vi hoạt động.
2. Cho mảng sau đây $A = \{7, 4, 8, 3, 10, 2\}$ Sau đó:
 - a. Hãy liệt kê các phần tử của mảng segment tree sau khi thực hiện build xong cây segment tree thực hiện truy vấn tìm max
Các phần tử sau khi thực hiện build xong cây segment tree thực hiện truy vấn tìm max là: 10 , 8 , 10 , 7 , 8 , 10 , 2 , 7 , 4 , 8 , 3.
 - b. Vẽ cây segment tree ở trên



- c. Nội ý tưởng của hàm truy vấn tìm max của cây segment tree trên
Bắt đầu với $left = 0, right = n-1$;
- Bước 1: nếu $left$ khác $right$ thì:
 - Thực hiện hàm đệ quy với $right = \lfloor right/2 \rfloor$
 - Thực hiện hàm đệ quy với $left = \lfloor right/2 \rfloor + 1$
 - Nếu MAX bên trái lớn hơn bên phải thì trả về MAX bên trái
 - Ngược lại thì trả về MAX bên phải
 - Quay về bước 1
 - Bước 2: Ngược lại nếu $left$ bằng $right$ thì trả về giá trị tại đó. Quay lại bước 1.

Pseudo code:

```

Int MAX(Arr , segm_tree , left , right , i , j ){
    If ( left == right )
        Return Arr[left];
    Else{
        Int a = MAX (Arr , segm_tree , left ,  $\lfloor right/2 \rfloor$ );
        Int b = MAX (Arr , segm_tree ,  $\lfloor right/2 \rfloor + 1$  , right);
        Return ( a > b )? a : b;
    }
}

```

- d. Liệt kê các phần tử của mảng segment tree sau khi thực hiện update $A[3]=12$
- Các phần tử trong mảng sau khi thực hiện cập nhật segment tree là:
12 , 12 , 10 , 7 , 12 , 10 , 2 , 7 , 4 , 12 , 3