

Lazy Propagation in Segment Tree

CS523.011 - Group 6

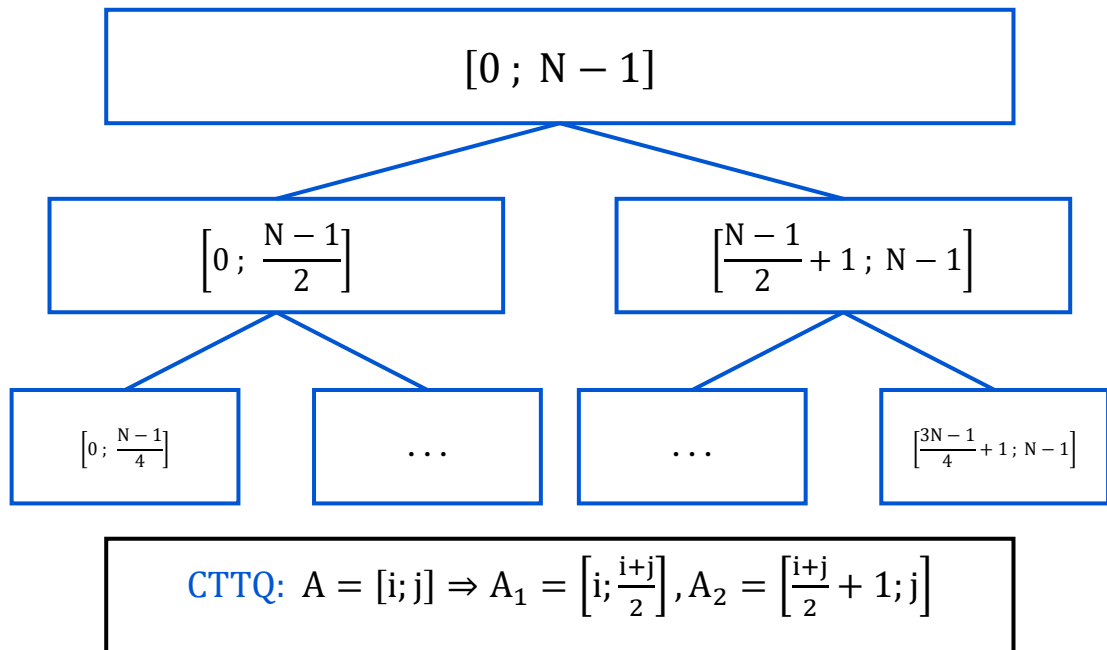
Nguyễn Hoàng Xuân Huy
21520924

Trần Ngọc Thiện
21521465

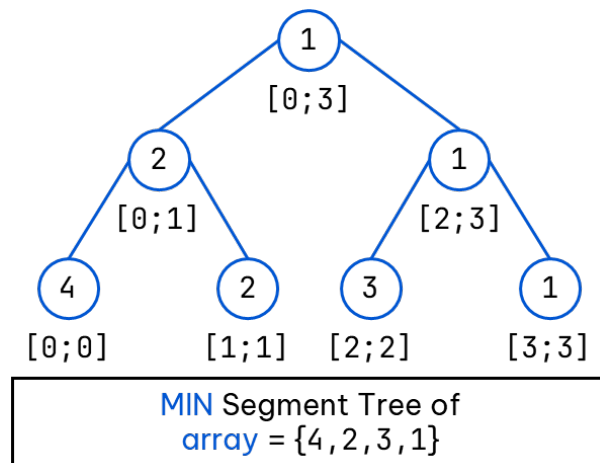
1. Introduction

a) What is Segment Tree?

- » Segment Tree (Cây Phân Đoạn) là 1 CTDL dạng cây, thường được sử dụng trong các bài toán cần xử lý trên array.
- » Segment Tree được xây dựng dựa trên phương pháp “Chia để Trị” với mỗi nút trên cây quản lý một đoạn trên array:



- » Các nút này chứa dữ liệu về kết quả của một phép toán hay phép truy vấn nào đó trên đoạn mà nó quản lý, ví dụ như:
 - > Cộng, Trừ, Nhân, Chia, ...
 - > Tìm Max, tìm Min, GCD, LCM, ...
 - > AND, OR, XOR, ...



b) Segment Tree Implementation (C++)

Segment Tree thường được cài đặt bằng array với 2 cách cài đặt: **đệ quy** và **không đệ quy**. Sau đây là ví dụ về 2 cách cài đặt trên cho Sum Segment Tree:

» **Khai báo:**

Cách 1	Cách 2
<pre>#include <bits/stdc++.h> using namespace std; int* arr = NULL; int* ST = NULL; int n; int main() { cin >> n; // Cấp phát động int *arr = new int[n]; int *ST = new int[n*4]; // Giải phóng bộ nhớ delete[] arr; delete[] ST; }</pre>	<pre>#include <bits/stdc++.h> using namespace std; int* arr = NULL; int* ST = NULL; int n; int main() { cin >> n; // Cấp phát động int *arr = new int[n]; int *ST = new int[n*2]; // Giải phóng bộ nhớ delete[] arr; delete[] ST; }</pre>

» **Nhập dữ liệu cho `arr` (optional):**

```
void InputArr() {
    for (int i=0; i<n; i++)
        cin >> arr[i];
}
```

» **Build Segment Tree**

Cách 1	<pre>void Build(int node, int l, int r) { if (l == r) ST[node] = A[l]; else { // Chia arr làm 2 phần int mid = (l+r) / 2; // Build đệ quy từ phần bên trái Build(node*2 + 1, l, mid); // Build đệ quy từ phần bên phải Build(node*2 + 2, mid+1, r); ST[node] = ST[node*2 + 1] + ST[node*2 + 2]; // sum } }</pre>
---------------	---

Giải thích:

- > $ST[0]$ nằm ở trên đỉnh là tổng của cả đoạn $[0; n-1]$
- > Node con của node thứ i là $(2i+1)$ và $(2i+2)$
VD: node con của node root là $ST[1]$ và $ST[2]$, node con của $ST[1]$ là $ST[3]$ và $ST[4]$, ...
- > [Example1.pptx](#)

Cách 2	<pre>void Build() { for (int i = 0; i < n; i++) // Gán node lá ST[n-1+i] = arr[i]; for (int i = n-2; i >= 0; i--) // Tính các node cha ST[i] = ST[i*2 + 1] + ST[i*2 + 2]; }</pre>
-------------------	---

Giải thích: Tương tự như trên:

- > $ST[0]$ nằm ở trên đỉnh là tổng của cả đoạn
- > Node con của node thứ i là $(2i+1)$ và $(2i+2)$
- > [Example2.pptx](#)

» **Query [L;R]**

Cách 1	<pre>void Query(int node, int tl, int tr, int l, int r) { if (r < tl tr < l) return 0; // Out of range if (l <= tl && tr <= r) return SR[node]; int tm = (tl+tr) / 2; return query(node*2 + 1, tl, tm, l, r) + query(node*2 + 2, tm+1, tr, l, r); }</pre>
-------------------	--

c) Segment Tree Implementation (Python)
