

Segment Tree & Deque

GV: TS. Phạm Quang Dũng
TA: Lê Minh Quang

Ngày 14 tháng 5 năm 2020

1 Segment Tree

- Phát biểu bài toán
- Tổng quát về Segment Tree
- Các thao tác

2 Deque

- Deque là gì?
- Định nghĩa stack đặc trưng của dãy
- Các thao tác với dãy

1 Segment Tree

- Phát biểu bài toán
- Tổng quát về Segment Tree
- Các thao tác

2 Deque

- Deque là gì?
- Định nghĩa stack đặc trưng của dãy
- Các thao tác với dãy

1 Segment Tree

- Phát biểu bài toán
- Tổng quát về Segment Tree
- Các thao tác

2 Deque

- Deque là gì?
- Định nghĩa stack đặc trưng của dãy
- Các thao tác với dãy

Phát biểu bài toán

Bài toán

Cho một dãy a có N phần tử, đánh số lần lượt là a_1, a_2, \dots, a_N . Cần thực hiện Q thao tác, mỗi thao tác thuộc 1 trong 2 loại:

- **Loại 1:** Gán giá trị v cho phần tử có chỉ số u , $a_u = v$.
- **Loại 2:** Tìm giá trị lớn nhất trong đoạn $[l, r]$.

Bài toán

Cho một dãy a có N phần tử, đánh số lần lượt là a_1, a_2, \dots, a_N . Cần thực hiện Q thao tác, mỗi thao tác thuộc 1 trong 2 loại:

- **Loại 1:** Gán giá trị v cho phần tử có chỉ số u , $a_u = v$.
- **Loại 2:** Tìm giá trị lớn nhất trong đoạn $[l, r]$.

Thuật toán đơn giản: sử dụng duy nhất mảng a để duy trì các giá trị.

- **Loại 1:** Đơn giản ta gán $a_u = v$. Độ phức tạp $O(1)$.
- **Loại 2:** Duyệt lần lượt các phần tử từ l đến r , tìm giá trị lớn nhất trong đoạn $[l, r]$. Độ phức tạp $O(N)$.

Vậy độ phức tạp tổng của thuật toán này là $O(NQ)$. Cần một thuật toán tốt hơn.

1 Segment Tree

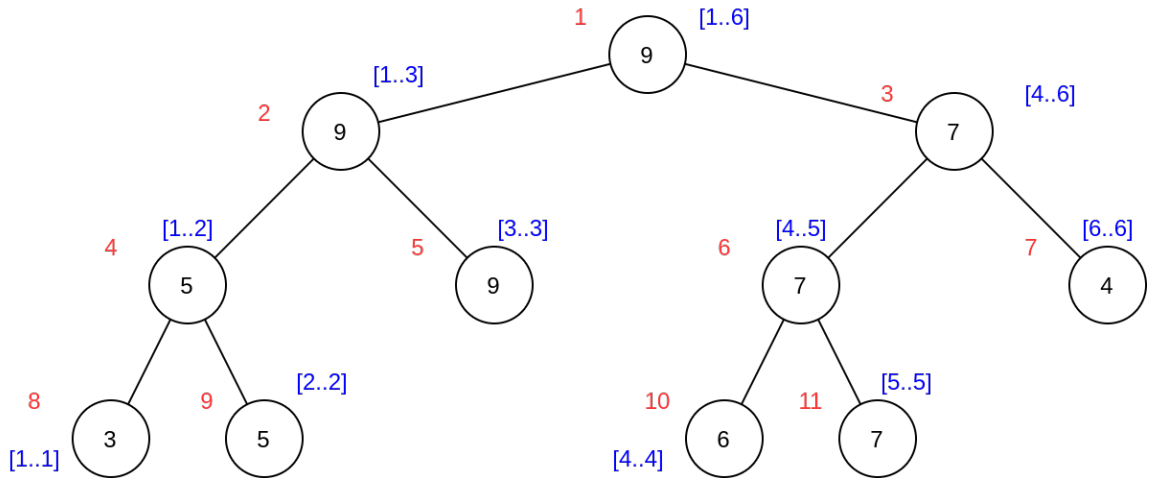
- Phát biểu bài toán
- Tổng quát về Segment Tree
- Các thao tác

2 Deque

- Deque là gì?
- Định nghĩa stack đặc trưng của dãy
- Các thao tác với dãy

- Segment Tree là 1 cây nhị phân đầy đủ.
- Mỗi nút quản lý một đoạn liên tiếp trên dãy số.
- Nút gốc của cây sẽ quản lý cả dãy từ 1 đến N , nút con trái của nó sẽ quản lý đoạn $[1, N/2]$, nút con phải sẽ quản lý $[N/2 + 1, N]$.
- Một nút quản lý đoạn $[i, j]$, thì 2 nút con của nó quản lý $[i, (i+j)/2]$ và $[(i+j)/2 + 1, j]$.
- Mỗi nút lá quản lý duy nhất 1 phần tử.

Hình ảnh Segment Tree

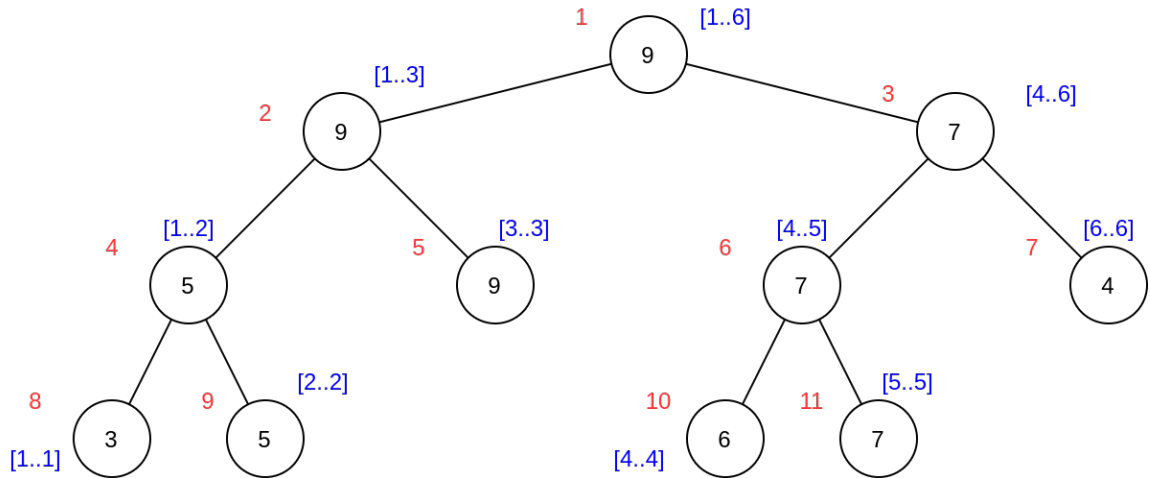


Mảng $A = \{3, 5, 9, 6, 7, 4\}$

Biểu diễn Segment Tree

- Có nhiều cách biểu diễn, đơn giản nhất có thể dùng 1 mảng 1 chiều.
- Phần tử có chỉ số 1 biểu diễn gốc cây.
- Phần tử có chỉ số id sẽ có 2 con: con trái là $id * 2$, con phải là $id * 2 + 1$.
- Mỗi nút sẽ lưu một giá trị nó quản lý, cụ thể trong bài toán trên, ta lưu một giá trị $maxVal$ là giá trị tối đa của tất cả phần tử nó quản lý.
- Khai báo mảng có kích thước $4 * N$.

Biểu diễn Segment Tree



Mảng $A = \{3, 5, 9, 6, 7, 4\}$

1 Segment Tree

- Phát biểu bài toán
- Tổng quát về Segment Tree
- Các thao tác

2 Deque

- Deque là gì?
- Định nghĩa stack đặc trưng của dãy
- Các thao tác với dãy

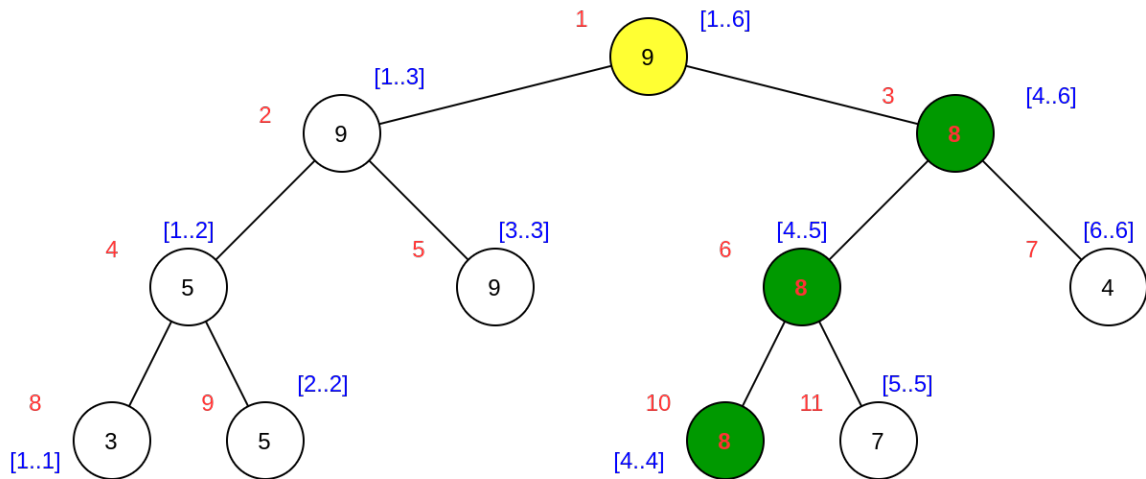
Nhập u, v , gán $a_u = v$

Cần cập nhật thông tin của tất cả các nút trên cây quản lý phần tử u này.

Các bước cập nhật:

- Gốc cây quản lý tất cả phần tử, nên ta đi thăm từ gốc xuống.
- Khi ta thăm 1 nút k , ta đảm bảo rằng k có quản lý phần tử a_u :
 - Nếu k là lá: nút k này chỉ quản lý 1 phần tử duy nhất a_u . Chúng ta chỉ đơn giản cập nhật $k.maxVal = v$.
 - Nếu k không à lá: ta cần xác định nút con của nó cũng quản lý a_u . Rõ ràng chỉ có 1 trong 2 nút con quản lý a_u . Sau đó, đi thăm nút con đó, cập nhật giá trị của nút con. Cuối cùng, cập nhật $k.maxVal = \max(k.l.maxVal, k.r.maxVal)$.
- Độ phức tạp thuật toán $O(\log N)$

Loại 1



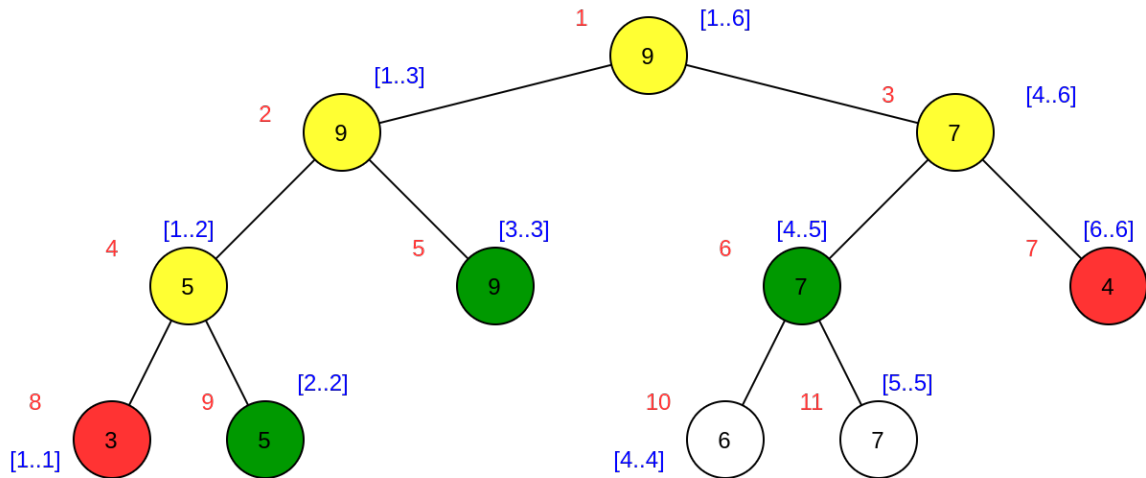
Mảng A = {3, 5, 9, 6->8, 7, 4}

Nhập l, r , cần tìm $\max_{i=l}^r a_i$

Cần tìm các nút trên cây quản lý đoạn nằm hoàn toàn trong $[l, r]$, lấy max giá trị $maxVal$ của các đoạn đó.

Các bước truy vấn:

- Ta đi thăm các nút từ gốc xuống dưới.
- Khi thăm nút k , giả sử nút này quản lý đoạn $[x, y]$. Ta xem xét các trường hợp sau:
 - $[l, r]$ và $[x, y]$ không giao nhau ($x > r \vee y < l$). Đoạn mà k quản lý không liên quan đến đoạn cần truy vấn. Ta rời khỏi hàm ngay lập tức mà không cần thăm các con của k .
 - $[l, r]$ chứa hoàn toàn $[x, y]$ ($l \leq x \wedge y \leq r$). Kết quả chính là $k.maxVal$. Chúng ta không cần thăm 2 con của nó.
 - $[l, r]$ giao với $[x, y]$. Ta không thể lấy luôn $k.maxVal$ làm kết quả vì $[l, r]$ chứa 1 vài phần tử không có trong $[x, y]$. Vì thế ta đi thăm 2 con của k và lấy kết quả là max của kết quả 2 con.
- Độ phức tạp thuật toán $O(\log N)$.



Truy vấn max trên khoảng $[2, 5]$.

1 Segment Tree

- Phát biểu bài toán
- Tổng quát về Segment Tree
- Các thao tác

2 Deque

- Deque là gì?
- Định nghĩa stack đặc trưng của dãy
- Các thao tác với dãy

1 Segment Tree

- Phát biểu bài toán
- Tổng quát về Segment Tree
- Các thao tác

2 Deque

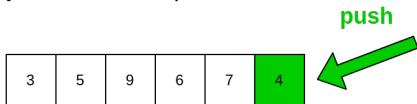
- Deque là gì?
- Định nghĩa stack đặc trưng của dãy
- Các thao tác với dãy

Deque là gì?

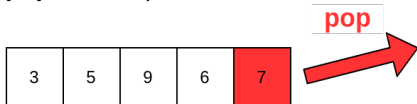
Stack

3	5	9	6	7
---	---	---	---	---

- Là một cấu trúc có tính chất vào trước ra sau.
- Nó hỗ trợ 2 thao tác:
 - **push**: Thêm 1 phần tử vào cuối stack.



- **pop**: Loại 1 phần tử ở cuối stack.

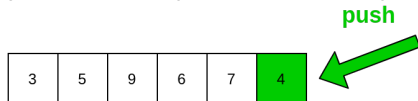


Deque là gì?

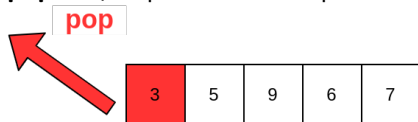
Queue

3	5	9	6	7
---	---	---	---	---

- Là một cấu trúc có tính chất vào trước ra trước.
- Nó hỗ trợ 2 thao tác:
 - **push**: Thêm 1 phần tử vào cuối queue.



- **pop**: Loại 1 phần tử ở đầu queue.



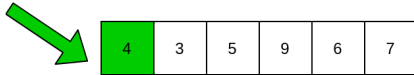
Deque là gì?

Deque

3	5	9	6	7
---	---	---	---	---

- Kết hợp và mở rộng 2 cấu trúc trên.
- Nó hỗ trợ 4 thao tác, tất cả được thực hiện trong $O(1)$.
 - **push_back**: Thêm 1 phần tử vào cuối deque.
 - **push_front**: Thêm 1 phần tử vào đầu deque.

push



- **pop_back**: Loại 1 phần tử ở cuối deque.
- **pop_front**: Loại 1 phần tử ở đầu deque.

1 Segment Tree

- Phát biểu bài toán
- Tổng quát về Segment Tree
- Các thao tác

2 Deque

- Deque là gì?
- Định nghĩa stack đặc trưng của dãy
- Các thao tác với dãy

Định nghĩa stack đặc trưng của dãy

Cho 1 dãy a có n phần tử, lần lượt là a_1, a_2, \dots, a_n .

- Stack chứa tất cả các phần tử mang giá trị lớn nhất trong tất cả các phần tử từ nó cho đến cuối dãy

3	5	9	6	7	4
---	---	---	---	---	---

3	5	6
---	---	---

Các tính chất của stack đặc trưng của dãy

Tính chất của stack này:

- Xét mỗi phần tử v trong stack, giá trị lớn nhất của các phần tử trong đoạn $[v, n]$ (với n là phần tử cuối cùng của dãy) là a_v .
- n luôn luôn xuất hiện ở cuối dãy.
- Phần tử lớn nhất của dãy là đầu của stack.
- Xác định duy nhất 1 stack ứng với 1 dãy.
- Sử dụng cấu trúc **deque** để lưu trữ stack này.

3	5	9	6	7	4
---	---	---	---	---	---

3	5	6
---	---	---

1 Segment Tree

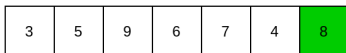
- Phát biểu bài toán
- Tổng quát về Segment Tree
- Các thao tác

2 Deque

- Deque là gì?
- Định nghĩa stack đặc trưng của dãy
- Các thao tác với dãy

Thêm 1 phần tử vào x cuối dãy

- Giả sử ta đã dựng được stack của dãy cũ.
- Vì đỉnh stack luôn luôn là phần tử cuối cùng, ta biết phần tử mới thêm sẽ ở đỉnh stack.
- Cần tìm phần tử gần nhất lớn hơn nó.
- Xét đỉnh stack, nếu đỉnh stack mang giá trị $> x$, ta dừng lại, bởi đây là giá trị cần tìm. Ngược lại, khi nó $\leq x$, ta loại nó ra khỏi stack, bởi nó không phải là giá trị cần tìm.



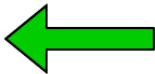
Độ phức tạp thuật toán: trung bình $O(1)$ khi thêm phần tử.

Ví dụ thêm phần tử

Mảng a

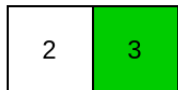


Stack



Ví dụ thêm phần tử

Mảng a

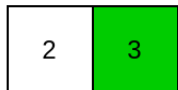


Stack

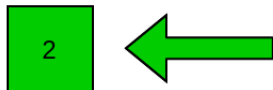


Ví dụ thêm phần tử

Mảng a



Stack



Ví dụ thêm phần tử

Mảng a

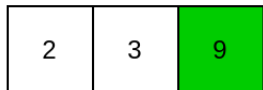
2	3	9
---	---	---

Stack

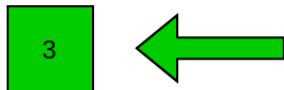


Ví dụ thêm phần tử

Mảng a



Stack

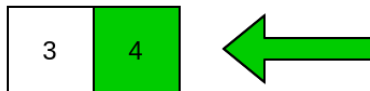


Ví dụ thêm phần tử

Mảng a

2	3	9	6
---	---	---	---

Stack

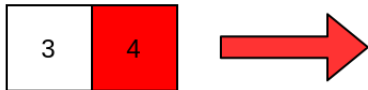


Ví dụ thêm phần tử

Mảng a

2	3	9	6	7
---	---	---	---	---

Stack

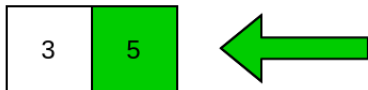


Ví dụ thêm phần tử

Mảng a

2	3	9	6	7
---	---	---	---	---

Stack




Ví dụ thêm phần tử

Mảng a

2	3	9	6	7	4
---	---	---	---	---	---

Stack

3	5	6
---	---	---



Ví dụ thêm phần tử

Mảng a

2	3	9	6	7	4	8
---	---	---	---	---	---	---

Stack

3	5	6
---	---	---



Ví dụ thêm phần tử

Mảng a

2	3	9	6	7	4	8
---	---	---	---	---	---	---

Stack

3	5
---	---




Ví dụ thêm phần tử

Mảng a

2	3	9	6	7	4	8
---	---	---	---	---	---	---

Stack

3	7
---	---




Ví dụ thêm phần tử

Mảng a

2	3	9	6	7	4	8	5
---	---	---	---	---	---	---	---

Stack

3	7	8
---	---	---



Loại 1 phần tử ở đầu dãy

- Giả sử ta đã dựng được stack của dãy cũ.
- Nếu phần tử đầu là đáy của stack, loại bỏ nó.
- Nếu phần tử đầu không là đáy stack, bỏ qua.

10	5	9	6	7	4
----	---	---	---	---	---

1	3	5	6
---	---	---	---

Độ phức tạp thuật toán: $O(1)$ khi loại phần tử.