Organization: https://oj.vnoi.info/organization/294-tkn

Contest: <https://oj.vnoi.info/contest/tkn_st_stla_a2>

* Lý thuyết Segment Tree:
* Merge 2 nút con -> nút cha
* Modify (id,v) : cập nhật nút id với giá trị v
* Get(L,R) : giá trị của đoạn con L -> R
* Cấu trúc của mỗi nút
* Điều kiện để dùng Segment Tree: tính chất giao hoán : merge(merge(a,b),c) = merge(a,merge(b,c)) (tương tự như (a+b) + c = a + (b + c) )
* Tính chất quan trọng nhất của Segment Tree là số tầng của Segment Tree = khoảng log(n)

1. Bài LeftFirstX: Walk on Segment Tree ( Binary Search on Segment Tree)

* Nhận xét là với j nhỏ nhất sao cho a[j] >= x tương đương với tìm j nhỏ nhất sao cho max(L,j) >= x.
* Gọi F(x) là max( a[L],a[L+1],...,a[x] ). Ta thấy rằng F(x) <= F(x+1) => có tính chất chặt nhị phân
* Thuật O(n\*log(n)^2): chặt nhị phân, dùng segtree để lấy max
* Thuật O(n\*log(n)) : <https://leduythuccs.github.io/2020-07-10-Binary-Search-on-Segment-Tree/>
* Code: <https://ideone.com/o4cE5B>
* Persistent Segment Tree
* Lưu tất cả các segment tree sau q lần update
* Nhận xét: sau mỗi lần update có tối đa log(n) nút bị thay đổi => thêm log(n) nút với giá trị được update vào cây sau update
* Khi query chỉ cần quan tâm đến root của cây
* Cấu trúc của nút trên cây sẽ thêm 2 giá trị idL là chỉ số của nút con trái, idR là chỉ số của nút con phải, khi cập nhật / truy vấn sẽ đi theo idL và idR

Contest: <https://oj.vnoi.info/contest/tkn_stadvanced_a2>

1. Anh nhân viên chăm chỉ

* Với mỗi phần tử a[i] lưu thành 1 pair:

+ a[i].first = value (giá trị)

+ a[i].second = i (chỉ số)

* Sort lại mảng a
* Nếu chỉ xét t phần tử đầu của mảng a, thì với mỗi query(i,j,k), giá trị đáp án của query là a[t].first với giá trị t đầu tiên mà xét t phần tử đầu của mảng A thì số lượng phần tử trên đoạn có giá trị second từ i -> j >= k
* Code Persistent Segment Tree: <https://ideone.com/ZMOfSg> (không phải cho bài này)

+ Cntnw = số lượng nút hiện tại

Struct tk

{

Lf: chỉ số của nút con bên trái

Rt: chỉ số của nút con bên phải

}

+ Hàm update sẽ trả về chỉ số mới của các nút khi sử dụng PST

* Chặt nhị phân từ 1 -> n, với giá trị mid tạo hàm check(mid) để kiểm tra nếu xét mid giá trị đầu thì số lượng phần tử có giá trị second từ i -> j >= k
* Dùng PST lưu n trạng thái của ST, trạng thái i là sau khi ta update i phần tử đầu trên mảng a
* Code bài: https://ideone.com/8lxcVl