

Ngày 02 tháng 3 năm 2021

# BINARY INDEXED TREE [BIT]

## I-LÝ THUYẾT

Cây xuôi	Cây ngược
<pre> // Cây tổng struct SUMLBIT {     int bit[maxn];      void Init() {         for (int i = 1; i &lt;= n; ++i)             bit[i] = 0;     }      // Cap nhat a[u] += val     void Update (int u, int val) {         while (u &lt;= n) {             bit[u] += val;             u += u &amp; (-u);         }     }      // Tinh a[1]+a[2]+...+a[u]     int Get (int u) {         int kq = 0;         while (u &gt; 0) {             kq += bit[u];             u -= u &amp; (-u);         }         return kq;     } }; </pre>	<pre> // Cây tổng struct SUMRBIT {     int bit[maxn];      void Init() {         for (int i = 1; i &lt;= n; ++i)             bit[i] = 0;     }      // Cap nhat a[u] += val     void Update (int u, int val) {         while (u &gt; 0) {             bit[u] += val;             u -= u &amp; (-u);         }     }      // Tinh a[u]+a[u+1]+...+a[n]     int Get (int u) {         int kq = 0;         while (u &lt;= n) {             kq += bit[u];             u += u &amp; (-u);         }         return kq;     } }; </pre>
<pre> // Cây Max struct MAXLBIT {     int bit[maxn];      void Init() {         for (int i = 1; i &lt;= n; ++i)             bit[i] = -INF;     }      // Cap nhat a[u] = val (&gt; GT cu a[u])     void Update (int u, int val) {         while (u &lt;= n) {             bit[u] = max(bit[u], val);             u += u &amp; (-u);         }     }      // Tinh max(a[1],a[2],...,a[u])     int Get (int u) {         int kq = -INF;         while (u &gt; 0) {             kq = max(kq, bit[u]);             u -= u &amp; (-u);         }         return kq;     } }; </pre>	<pre> // Cây Max struct MAXRBIT {     int bit[maxn];      void Init() {         for (int i = 1; i &lt;= n; ++i)             bit[i] = 0;     }      // Cap nhat a[u] = val (&gt; GT cu a[u])     void Update (int u, int val) {         while (u &gt; 0) {             bit[u] = max(bit[u], val);             u -= u &amp; (-u);         }     }      // Tinh max(a[u], a[u+1], ...,a[n])     int Get (int u) {         int kq = 0;         while (u &lt;= n) {             kq += bit[u];             u += u &amp; (-u);         }         return kq;     } }; </pre>

## II-BÀI TẬP

### A. Truy vấn tổng [SUMBIT]

Cho mảng  $A = (a_1, a_2, \dots, a_n)$  các số nguyên. Hãy thực hiện  $m$  truy vấn trên mảng này, mỗi truy vấn thuộc một trong ba loại:

- **1 u  $\Delta$**  : Tăng phần tử  $a_u$  lên một lượng  $\Delta$  ( $|\Delta| \leq 10^9$ )
- **2 u V** : Gán  $a_u = V$  ( $|V| \leq 10^9$ )
- **3 u v** : Tính tổng  $a_u + a_{u+1} + \dots + a_v$

**Input:**

- Dòng đầu tiên ghi số nguyên dương  $n$  ( $n \leq 10^5$ )
- Dòng thứ hai ghi  $n$  số nguyên  $a_1, a_2, \dots, a_n$  - mảng ban đầu. ( $|a_i| \leq 10^9$ )
- Dòng thứ ba ghi số nguyên dương  $m$  ( $1 \leq m \leq 10^5$ )
- $m$  dòng cuối cùng mỗi dòng ghi một truy vấn theo qui cách mô tả ở trên

**Output:** Với các truy vấn loại 3 in ra kết quả tìm được trên một dòng

**Example:**

Input	Output
5	13
1 2 3 4 5	10
6	
1 4 3	
2 2 3	
3 2 4	
1 5 2	
2 4 -4	
3 1 5	

### B. Truy vấn max xuôi [MAXLBIT]

Cho mảng  $A = (a_1, a_2, \dots, a_n)$  các số nguyên. Hãy thực hiện  $m$  truy vấn trên mảng này, mỗi truy vấn thuộc một trong bốn loại:

- **1 u  $\Delta$**  : Tăng phần tử  $a_u$  lên một lượng  $\Delta$  ( $0 \leq \Delta \leq 10^4$ )
- **2 u** : Tính  $\max\{a_1, a_2, \dots, a_u\}$

**Input:**

- Dòng đầu tiên ghi số nguyên dương  $n$  ( $n \leq 10^5$ )
- Dòng thứ hai ghi  $n$  số nguyên  $a_1, a_2, \dots, a_n$  - mảng ban đầu. ( $|a_i| \leq 10^4$ )
- Dòng thứ ba ghi số nguyên dương  $m$  ( $1 \leq m \leq 10^5$ )
- $m$  dòng cuối cùng mỗi dòng ghi một truy vấn theo qui cách mô tả ở trên

**Output:** Với các truy vấn loại 2 in ra kết quả tìm được trên một dòng

**Example:**

Input	Output
5	3
1 2 3 4 5	10
5	10
1 5 3	
2 3	
1 3 7	
2 5	
2 4	

### C. Truy vấn max ngược [MAXRBIT]

Cho mảng  $A = (a_1, a_2, \dots, a_n)$  các số nguyên. Hãy thực hiện  $m$  truy vấn trên mảng này, mỗi truy vấn thuộc một trong bốn loại:

- **1 u  $\Delta$**  : Tăng phần tử  $a_u$  lên một lượng  $\Delta$  ( $0 \leq \Delta \leq 10^4$ )
- **2 u** : Tính  $\max\{a_u, a_{u+1}, \dots, a_n\}$

**Input:**

- Dòng đầu tiên ghi số nguyên dương  $n$  ( $n \leq 10^5$ )
- Dòng thứ hai ghi  $n$  số nguyên  $a_1, a_2, \dots, a_n$  - mảng ban đầu. ( $|a_i| \leq 10^4$ )
- Dòng thứ ba ghi số nguyên dương  $m$  ( $1 \leq m \leq 10^5$ )
- $m$  dòng cuối cùng mỗi dòng ghi một truy vấn theo qui cách mô tả ở trên

**Output:** Với các truy vấn loại 2 in ra kết quả tìm được trên một dòng

**Example:**

Input	Output
5	8
1 2 3 4 5	10
5	8
1 5 3	
2 5	
1 3 7	
2 3	
2 4	

**D. Tập hợp [ORDSET]**

Bạn cần quản lý một tập hợp động  $S$  hỗ trợ hai thao tác cơ bản:

- **INSERT(S,x)**: Nếu  $x$  không thuộc  $S$ , thêm  $x$  vào  $S$
- **DELETE(S,x)**: Nếu  $x$  thuộc  $S$ , xóa  $x$  khỏi  $S$

và hai loại truy vấn:

- **K-TH(S)**: Trả về số bé thứ  $k$  của  $S$
- **COUNT(S,x)**: Đếm số lượng số thuộc  $S$  bé hơn  $x$

**Input:**

- Dòng đầu tiên ghi  $Q$  ( $1 \leq Q \leq 200000$ ) - số thao tác
- $Q$  dòng sau, đầu mỗi dòng chứa ký tự I, D, K hoặc C cho biết thao tác tương ứng là INSERT, DELETE, K-TH hay COUNT. Tiếp theo là một khoảng trắng và một số nguyên là tham số cho thao tác đó.

Nếu tham số là  $x$ , dữ liệu đảm bảo  $|x| \leq 10^9$ . Nếu tham số là chỉ số  $k$  dữ liệu đảm bảo  $1 \leq k \leq 10^9$

**Output:** Với mỗi truy vấn in kết quả trên một dòng. Với truy vấn K-TH, nếu  $k$  lớn hơn số phần tử của  $S$ , in ra 'invalid'

**Example:**

Input	Output
8	1
I -1	2
I -1	2
I 2	invalid
C 0	
K 2	
D -1	
K 1	
K 2	

**E. Dãy số [MWSEQ]**

Xét một dãy số nguyên, người ta định nghĩa trọng số của dãy số là số phần tử của dãy con tăng ngặt dài nhất gồm các phần tử liên tiếp của dãy. Ví dụ dãy (1,2,3,8,2,4,6,7,9,9,2) có trọng số là 5.

Cho dãy số nguyên  $A = (a_1, a_2, \dots, a_n)$ , bạn được phép xóa (nếu muốn) một dãy các phần tử liên tiếp trong  $A$ . Hãy tìm cách xóa để dãy còn lại có trọng số lớn nhất, cho biết trọng số của dãy còn lại.

Ví dụ nếu  $A = (5, 3, 4, 9, 2, 6, 6, 7, 1)$  cách tốt nhất là xóa đi 3 phần tử liên tiếp 9, 2, 6; còn lại dãy  $(5, 3, 4, 6, 7, 1)$  có trọng số là 4.

**Input:**

- Dòng đầu tiên chứa một số nguyên dương  $n \leq 200000$
- Dòng thứ hai chứa  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  ( $\forall i: a_i \leq 10^9$ ) cách nhau bởi dấu cách.

**Output:** Một số nguyên duy nhất là trọng số của dãy còn lại.

**Example:**

input	output
9 5 3 4 9 2 6 6 7 1	4

**Subtasks:**

- Subtask 1:  $n \leq 5000$
- Subtask 2:  $5000 < n \leq 10^5$

**F. Số nghịch thế [CINV]**

Cho dãy  $A = (a_1, a_2, \dots, a_n)$  một cặp  $(i, j): 1 \leq i < j \leq n$  được gọi là một nghịch thế nếu như:

- $i < j$
- $a_i > a_j$

Hãy đếm xem dãy  $A$  có bao nhiêu nghịch thế?

**Input:**

- Dòng đầu tiên ghi số nguyên dương  $n$
- Dòng thứ hai ghi  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $|a_i| \leq 10^9$ )

**Output:** Một số nguyên duy nhất là số lượng nghịch thế tìm được.

**Example:**

Input	Output
3 2 1 3	1

**G. Dãy con tăng tổng quát [INCWSEQ]**

Cho dãy số nguyên dương  $A = (a_1, a_2, \dots, a_n)$ , phần tử  $a_i$  có trọng số  $w_i$ . Một dãy  $(a_{i_1}, a_{i_2}, \dots, a_{i_k})$  thỏa mãn:

$$\begin{cases} 1 \leq i_1 < i_2 < \dots < i_k \leq n \\ a_{i_1} < a_{i_2} < \dots < a_{i_k} \end{cases}$$

được gọi là một dãy con tăng của dãy  $A$ . Chú ý rằng dãy chỉ gồm duy nhất một phần tử của  $A$  cũng được gọi là một dãy con tăng của  $A$ .

**Yêu cầu:** Trong các dãy con tăng của  $A$  hãy chỉ ra một dãy có tổng trọng số lớn nhất có thể.

**Input:**

- Dòng 1 chứa số nguyên dương  $n \leq 10^5$
- Dòng 2 chứa  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  theo đúng thứ tự đó ( $\forall i: a_i \leq 10^5$ )
- Dòng 3 chứa  $n$  số nguyên dương  $w_1, w_2, \dots, w_n$  theo đúng thứ tự đó ( $\forall i: w_i \leq 10^9$ )

**Output:**

- Dòng 1 ghi số phần tử của dãy con tìm được ( $m$ )
- Dòng 2 ghi  $m$  chỉ số của các phần tử được chọn theo thứ tự tăng dần

Các số trên một dòng của Input/Output được/phải ghi cách nhau ít nhất một dấu cách.

**Example:**

input	output
10 1 2 3 6 4 5 9 6 7 8 11 22 33 66 44 55 999 66 77 88	6 1 2 3 5 6 7

## H. Chuỗi ngọc [PEARLS]

Dọc theo bờ biển của hoang đảo có  $n$  viên ngọc trai do bọn cướp biển để lại, các viên ngọc đánh số  $1, 2, \dots, n$ . Viên thứ  $i$  có kích thước  $a_i$  và giá trị là  $b_i$ . Trước khi theo chim thần trở về đất liền, người em (trong truyện Cây Khế) quyết định nhặt các viên ngọc trai để xâu thành chuỗi ngọc tặng cô bạn gái xinh đẹp làng bên. Do thời gian gấp nên mỗi lần gặp một viên ngọc anh ta có ba sự lựa chọn:

1. Bỏ qua viên ngọc này
2. Nếu kích thước viên ngọc lớn hơn kích thước của viên ngọc bên phải chuỗi hạt trên tay, anh ta có thể nhặt viên ngọc này và thêm vào bên phải chuỗi hạt.
3. Nếu kích thước của viên ngọc nhỏ hơn kích thước của viên ngọc bên trái chuỗi hạt trên tay, anh ta có thể nhặt viên ngọc này và thêm vào bên trái chuỗi hạt.

Khởi đầu trên tay người em không có viên ngọc nào và khi ngồi trên lưng chim, anh ta có một chuỗi hạt với kích thước tăng dần tính từ trái qua phải.

**Yêu cầu:** Hãy giúp người em chọn chuỗi ngọc sao cho tổng giá trị của nó là lớn nhất. Chương trình chỉ in ra tổng giá trị này.

### Input:

- Dòng đầu tiên ghi số nguyên dương  $n$  ( $n \leq 10^5$ )
- Dòng 2 ghi dãy  $a_1, a_2, \dots, a_n$  ( $|a_i| \leq 10^9$ )
- Dòng 3 ghi dãy  $b_1, b_2, \dots, b_n$  ( $|b_i| \leq 10^9$ )

**Output:** In ra một số nguyên là tổng giá trị lớn nhất của chuỗi ngọc.

### Example:

Input	Output
10 1 2 3 6 4 5 9 6 7 8 11 22 33 66 44 55 999 66 77 88	1164

### Subtasks:

- Subtask 1:  $n \leq 5000$
- Subtask 2:  $n \leq 10^5, a_i \leq 10^5$
- Subtask 3:  $n \leq 10^5$