HƯỚNG DẪN GIẢI BÀI TẬP BINARY SREACH

BTBS02: Tìm vị trí

Yêu cầu: Cho mảng A có n phần tử số nguyên và một số nguyên x. Hãy lập trình thực hiện các yêu cầu sau:

- Sắp xếp mảng A tăng dần
- Tìm vị trí đầu tiên của phần tử x trong mảng A
- Tìm vị trí cuối cũng của phần tử x trong mảng A
- Tìm vị trí đầu tiên của phần tử lớn hơn x trong mảng A

Dữ liệu:

- Dòng đầu cho 2 số nguyên n và x (0 < n <= 10^7)
- Dòng tiếp theo chứa n số nguyên cách nhau khoảng trắng là giá trị các phần tử trong mảng A.
- x, và Ai có giá trị trong miền giá trị kiểu int

Kết quả: Ghi ra các vị trí tìm kiếm theo yêu cầu, mỗi vị trí trên một dòng theo định dạng như testcase mẫu. Trường hợp không tìm thấy thì dòng đó ghi "No"

Ví dụ:

Dữ liệu	Kết quả
10 3 1 1 2 2 3 3 3 5 7 9	Vi tri dau tien cua 3 trong mang A la 5 Vi tri cuoi cung cua 3 trong mang A la 7 Vi tri cua phan tu dau tien > 3 trong mang A la 8, gia tri 5

Hướng dẫn giải:

Nhận xét: Binary sreach cải tiến

Cấu trúc dữ liệu:

```
const int nmax = 1e7+5;
int a[nmax], n, x, vt;
```

- Mảng số nguyên A lưu dữ liệu đầu vào
- Biến N lưu số lượng phần tử của mảng
- Biến X lưu giá trị phần tử cần tìm
- Biến VT lưu vị trí phần tử X tìm được trong mảng A theo yêu cầu đề bài.

Giải thuật:

- Tìm vị trí đầu tiên của X trong mảng A: kế thừa từ thuật toán tìm kiếm nhị phân của phần tử X gặp đầu tiên trong mảng A.
  - Bổ sung thêm 1 biến VT
  - Ban đầu VT = -1
  - Nếu A[mid] == X thì
    - VT = mid
    - Vì ta cần tìm phần tử tận cùng bên trái trong mảng A có giá trị bằng X nên lúc này ta không thoát khỏi hàm tìm kiếm nhị phân mà sẽ giảm R = mid-1 sau đó tiếp tục thực hiện việc tìm kiếm cho đến khi mảng A không còn chia đôi được nữa (L > R)
- Tìm vị trí cuối cùng của X trong mảng A: kế thừa từ thuật toán tìm kiếm nhị phân tìm phần tử X đầu tiên trong mảng A tuy nhiên sau khi gặp phần tử có giá trị bằng X thì ta sẽ tăng L = mid+1 sau đó tiếp tục thực hiện việc tìm kiếm cho đến khi mảng A không còn chia đôi được nữa (L > R)

THẦY QUANG LUYỆN THI CHUYÊN TIN

- Tìm vị trí đầu tiên của phần tử lớn hơn x trong mảng A: kế thừa từ thuật toán tìm kiếm nhị phân tìm vị trí phần tử X đầu tiên trong mảng A tuy nhiên điều kiện để cập nhật vi trí là A[mid] <= X.
- Tìm vị trí đầu tiên của phần tử nhỏ hơn x trong mảng A: kế thừa từ thuật toán tìm kiếm nhị phân tìm vị trí đầu tiên của phần tử lớn hơn x trong mảng A
    - Tuy nhiên điều kiện để cập nhật vi trí là A[mid] >= X.
        - VT = mid
        - R = mid-1

	0	1	2	3	4	5	6	7	8	9	10
A	1	2	4	4	4	4	4	7	7	10	12

THẦY QUANG LUYỆN THI CHUYÊN TIN

BTBS03: Tìm phần tử

Yêu cầu: Cho hai dãy số nguyên $A = (a_1,a_2,...,a_m)$ và $B = (b_1,b_2,...,b_n)$. Với mỗi phần tử của dãy b_j của dãy B, hãy tìm phần tử a_i của dãy A gần b_i nhất. Nếu có nhiều hơn một phần tử thỏa mãn, hãy đưa ra phần tử a_i có giá trị nhỏ nhất.

Dữ liệu:

- Dòng 1 chứa hai số nguyên dương $m,n \le 10^6$
- Dòng 2 chứa m số nguyên $a_1, a_2, ..., a_m$ ($\forall i: |a_i| < 2.10^9$)
- Dòng 3 chứa n số nguyên $b_1, b_2, ..., b_n$ ($\forall j: |b_j| < 2.10^9$)

Kết quả: Gồm n dòng, dòng thứ j là kết quả tìm được của số b_j.

Ví dụ:

Dữ liệu	Kết quả
5 5 1 3 5 7 9 2 4 8 1 6	1 3 7 1 5

Hướng dẫn giải:

Nhận xét: Binary sreach cải tiến

Cấu trúc dữ liệu:

```
int a[nm], b[nm], m, n;
```

- Mảng số nguyên A, B lưu dữ liệu đầu vào
- Biến M, N lưu số lượng phần tử của mảng

Giải thuật:

- Nhận xét: Tìm phần tử a_i gần b_j nhất có nghĩa abs(A_i – B_j) là bé nhất.
- Cách 1:
  - Vì đề bài có nói nếu có nhiều phần tử A_i thõa mãn thì lấy A_i bé nhất do đó ta thực hiện tìm kiếm nhị phân để tìm ra vị trí (VT) phần tử A[i] đầu tiên lớn hơn Bj
  - Xét 3 trường hợp
    - Nếu VT = -1 (B_j lớn hơn tất cả các phần tử trong mảng A) lúc này phần tử trong A thõa mãn gần Bj nhất chính là A[m-1]
    - Nếu VT = 0 (phần tử nhỏ nhất trong mảng A bằng với B_j) lúc này A[0] chính là phần tử thõa mãn yêu cầu.
    - Xét độ chênh lệch giữa A[VT] – B_j và A[VT-1] – B_j, nếu giá trị nào nhỏ hơn thì xuất phần tử A tại vị trí đó.
- Cách 2:
  - Ban đầu giả sử phần tử A[0] là giá trị cần tìm thõa yêu cầu đề bài SO = A[0]
  - Ở mỗi bước tìm kiếm nhị phân, nếu chênh lệch giữa A[mid] và Bj nhỏ hơn hoặc bằng chênh lệch giữa SO và Bj và A[mid] nhỏ hơn SO thì ta cập nhật SO = A[mid]

	0	1	2	3	4
A	1	3	5	7	9
B	2	4	8	1	6

THẦY QUANG LUYỆN THI CHUYÊN TIN

BTBS04: Tìm max min

Yêu cầu: Cho hai dãy số nguyên $A = (a_1, a_2, ..., a_m)$ và $B = (b_1, b_2, ..., b_n)$. Với mỗi phần tử của dãy b_j của dãy B, hãy tìm hai chỉ số nhỏ nhất $imin$ và lớn nhất $imax$ sao cho $a_{imin} = a_{imax} = b_j$

Dữ liệu:

- Dòng 1 chứa hai số nguyên dương $m,n \le 10^6$
- Dòng 2 chứa m số nguyên $a_1, a_2, ..., a_m$ $(\forall i: |a_i| < 2.10^9)$
- Dòng 3 chứa n số nguyên $b_1, b_2, ..., b_n$ $(\forall j: |b_j| < 2.10^9)$

Kết quả: gồm n dòng, dòng thứ j ghi hai số $imin$ và $imax$ tìm được. Trong trường hợp không xuất hiện phần tử có giá trị bằng b_j trong dãy A ghi ra duy nhất số 0.

Ví dụ:

Dữ liệu	Kết quả
15 5 1 2 3 4 4 4 4 4 5 6 7 8 9 10 -2 0 4 9 12	0 0 4 9 14 14 0

Hướng dẫn giải:

Nhận xét: Binary sreach cải tiến

Cấu trúc dữ liệu:

```
int a[nmax], n, x, vt;
```

- Mảng số nguyên A lưu dữ liệu đầu vào
- Biến N lưu số lượng phần tử của mảng

Giải thuật:

- Kết hợp 2 ý tìm phần từ đầu tiên trong A có giá trị bằng B_j và tìm phần tử cuối cùng trong A có giá trị bằng B_j

THẦY QUANG LUYỆN THI CHUYÊN TIN