

[Sắp xếp - Tìm Kiếm]. Bài 33. Distinct Number

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Bạn được cung cấp một danh sách gồm n số nguyên và nhiệm vụ của bạn là tính số giá trị khác biệt trong danh sách.

Input Format

Input Dòng nhập đầu tiên có số nguyên n : số giá trị. Dòng thứ hai có n số nguyên x_1, x_2, \dots, x_n .

Constraints

$$1 \leq n \leq 2 \cdot 10^5; 1 \leq x_i \leq 10^9$$

Output Format

Output In một số nguyên: số lượng các giá trị riêng biệt.

Sample Input 0

```
5
1 2 3 3 2
```

Sample Output 0

```
3
```

[Comparator_Binary Search]. Bài 1. Comparator

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Bài 1. Cho mảng A[] gồm N phần tử, bạn hãy sắp xếp các phần tử trong mảng theo yêu cầu như sau :

- Sắp xếp số lượng chữ số chẵn xuất hiện trong số tăng dần, nếu 2 số còn cùng số lượng chữ số chẵn thì số nhỏ hơn in trước.
- Sắp xếp theo số lượng chữ số lẻ xuất hiện trong số ban đầu tăng dần, nếu 2 số có cùng số lượng chữ số lẻ thì số nào xuất hiện trước sẽ in ra trước. (Bài này các bạn làm theo mảng và vector để luyện tập thêm)

Input Format

- Dòng 1 là N : số lượng phần tử trong mảng
- Dòng 2 là N số trong mảng

Constraints

- $1 \leq N \leq 10^5$
- $0 \leq A[i] \leq 10^9$

Output Format

- Dòng 1 in ra các phần tử sau khi sắp xếp theo yêu cầu 1
- Dòng 2 in ra các phần tử sau khi sắp xếp theo yêu cầu 2

Sample Input 0

```
12
4 9 5 7 15 4 13 15 1 8 4 15
```

Sample Output 0

```
1 5 7 9 13 15 15 4 4 4 8
4 4 8 4 9 5 7 1 15 13 15 15
```

[Comparator_Binary Search]. Bài 2. Sắp xếp theo trị tuyệt đối

- [Problem](#)
- [Submissions](#)

- **Discussions**

Cho mảng A[] gồm N phần tử và số nguyên dương X, bạn hãy sắp xếp các phần tử trong mảng theo yêu cầu như sau :

- Sắp xếp các phần tử trong mảng theo abs của hiệu của các phần tử này với X tăng dần (tính trị tuyệt đối của A[i] với X rồi sắp tăng dần). Nếu có 2 phần tử có cùng giá trị sắp xếp này thì số nào nhỏ hơn sẽ in trước.
- Sắp xếp so cho các phần tử là số chẵn đứng trước, các phần tử là số lẻ đứng sau, trong đó chẵn tăng dần, lẻ giảm dần với 1 comparator duy nhất, không tách mảng.

Input Format

- Dòng 1 là N và X : số lượng phần tử trong mảng và số X
- Dòng 2 là N số trong mảng

Constraints

- $1 \leq N \leq 10^5$
- $0 \leq A[i], X \leq 10^9$

Output Format

- Dòng 1 in ra các phần tử sau khi sắp xếp theo yêu cầu 1
- Dòng 2 in ra các phần tử sau khi sắp xếp theo yêu cầu 2

Sample Input 0

```
14 26
83 86 49 15 31 81 33 10 77 38 49 86 71 28
```

Sample Output 0

```
28 31 33 15 38 10 49 49 71 77 81 83 86 86
10 28 38 86 86 83 81 77 71 49 49 33 31 15
```

[Comparator_Binary Search]. Bài 3. Lower_bound, Upper_bound

- **Problem**
- **Submissions**
- **Discussions**

Cho mảng A[] gồm N phần tử ĐÃ SẮP XẾP TĂNG DẦN và số nguyên X, nhiệm vụ của bạn là :

- 1. In ra chỉ số đầu tiên của phần tử $\geq X$ trong mảng, nếu không có phần tử nào $\geq X$ thì in ra -1.
 - 2. In ra chỉ số đầu tiên của phần tử $> X$ trong mảng, nếu không có phần tử nào $> X$ thì in ra - 1.
 - 3. In ra chỉ số đầu tiên của phần tử X trong mảng, nếu X không tồn tại in ra -1.
 - 4. In ra chỉ số cuối cùng của phần tử X trong mảng, nếu X không tồn tại in ra -1.
 - 5. Từ kết quả của 3 và 4 in ra số lần xuất hiện của X trong mảng.
- CHÚ Ý : Bạn code với 2 trường hợp, sử dụng mảng và vector, chỉ sử dụng hàm `lower_bound` và `upper_bound`

Input Format

- Dòng 1 là N : số lượng phần tử trong mảng
- Dòng 2 là N số trong mảng

Constraints

- $1 \leq N \leq 10^5$
- $0 \leq A[i] \leq 10^9$

Output Format

In ra 5 dòng kết quả của 5 yêu cầu

Sample Input 0

```
12 8
2 3 4 4 5 6 8 8 8 8 10 10
```

Sample Output 0

```
6
10
6
9
4
```

[Comparator_Binary Search]. Bài 4. Pair sort

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho mảng $A[]$ gồm N điểm trong hệ tọa độ Oxy, hãy sắp xếp các điểm này theo khoảng cách về gốc tọa độ tăng dần, nếu 2 điểm có cùng khoảng cách tới gốc tọa độ thì in ra theo hoành độ tăng dần, nếu tiếp tục 2 điểm này có cùng hoành độ thì in ra theo thứ tự tung độ tăng dần.

Input Format

- Dòng đầu tiên là N : số lượng điểm
- N dòng tiếp theo mỗi dòng là 2 số nguyên x, y tương ứng với hoành độ và tung độ

Constraints

- $1 \leq N \leq 10^5$;
- $-1000 \leq x, y \leq 1000$;

Output Format

In ra các điểm sau khi sắp xếp

Sample Input 0

```
13
```

```
-42 -76
47 43
12 7
-62 31
7 64
42 -92
-89 60
45 41
3 54
-41 40
20 -24
88 42
0 12
```

Sample Output 0

```
0 12
12 7
20 -24
3 54
-41 40
45 41
47 43
7 64
-62 31
-42 -76
88 42
42 -92
-89 60
```

[Comparator_Binary Search]. Bài 5. Sort pair 2

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho các điểm trong hệ tọa độ OXYZ, nhiệm vụ của bạn là sắp xếp các điểm này theo hoành độ tăng dần, nếu có cùng hoành độ thì sắp theo tung độ tăng dần, nếu có cùng tung độ tiếp tục sắp xếp theo cao độ tăng dần. Sử dụng pair như sau để lưu các tọa độ này :

```
pair<int, pair<int, int>>
```

Input Format

- Dòng đầu tiên là N : số lượng điểm.

- N dòng tiếp theo mỗi dòng là 3 số nguyên x, y, z tương ứng với 1 điểm

Constraints

- $1 \leq N \leq 10^5$
- x, y, z là số nguyên dương không quá 1000

Output Format

In ra các điểm sau khi sắp xếp

Sample Input 0

```
12
37 4 54
37 25 75
88 46 30
40 40 90
87 99 80
73 76 49
20 95 25
52 3 8
65 79 56
88 65 87
51 50 12
87 29 44
```

Sample Output 0

```
20 95 25
37 4 54
37 25 75
40 40 90
51 50 12
52 3 8
65 79 56
73 76 49
87 29 44
87 99 80
88 46 30
88 65 87
```

[Comparator Binary Search]. Bài 6. Sắp xếp lẻ

- Problem

- [Submissions](#)
- [Discussions](#)

Cho mảng A[] gồm N phần tử. Sắp xếp sao cho số nào có nhiều số lẻ hơn thì đứng trước, trong trường hợp nhiều số có cùng chữ số lẻ thì số nào nhỏ hơn sẽ đứng trước

Input Format

- Dòng 1 là N : số lượng phần tử trong mảng
- Dòng 2 là N số trong mảng

Constraints

- $1 \leq N \leq 10^6$
- $0 \leq A[i] \leq 10^9$

Output Format

In ra các số sau khi sắp xếp

Sample Input 0

```
11
4456 10493 23600 32757 6911 15236 14074 22458 32132 28596 15209
```

Sample Output 0

```
32757 6911 10493 15209 15236 32132 14074 28596 4456 22458 23600
```

[Comparator_Binary Search]. Bài 7. Sắp xếp 0, 6, 8

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho mảng A[] gồm N phần tử. Sắp xếp sao cho số có nhiều chữ số 0, 6 hoặc 8 hơn đứng trước, nếu có cùng số lượng chữ số 0 6 8 thì số nào nhỏ hơn sẽ đứng trước

Input Format

- Dòng 1 là N : số lượng phần tử trong mảng
- Dòng 2 là N số trong mảng

Constraints

- $1 \leq N \leq 10^6$
- $0 \leq A[i] \leq 10^9$

Output Format

In ra các số sau khi sắp xếp

Sample Input 0

```
14
25556 11105 32129 8765 25187 22479 21283 5842 4471 23648 26441 28433
23517 20475
```

Sample Output 0

```
8765 23648 5842 11105 20475 21283 25187 25556 26441 28433 4471 22479
23517 32129
```

[Comparator Binary Search]. Bài 8. Sắp xếp chữ số nguyên tố

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho mảng A[] gồm N phần tử. Sắp xếp sao cho số có nhiều chữ số nguyên tố hơn sẽ đứng trước, trong trường hợp các số có cùng số lượng chữ số nguyên tố thì số nào xuất hiện trước in ra trước

Input Format

- Dòng 1 là N : số lượng phần tử trong mảng
- Dòng 2 là N số trong mảng

Constraints

- $1 \leq N \leq 10^6$
- $0 \leq A[i] \leq 10^9$

Output Format

In ra các số sau khi sắp xếp

Sample Input 0

```
18
16501 5898 7388 26663 17330 15803 6865 3695 3495 20918 4244 580 29707
5410 11716 4765 12423 10073
```

Sample Output 0

```
17330 29707 12423 7388 26663 15803 3695 3495 4765 10073 16501 5898 6865
20918 4244 580 5410 11716
```

[Comparator Binary Search]. Bài 9. Sắp xếp pair theo second, first

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho mảng pair $A[]$ gồm N pair, sắp xếp mảng pair theo second tăng dần, nếu second bằng nhau thì sắp cho first giảm dần

Input Format

- Dòng 1 là N : số lượng phần tử trong mảng
- N dòng tiếp theo là N pair trong mảng

Constraints

- $1 \leq N \leq 10^6$
- Các phần tử của pair là số int

Output Format

In ra các pair sau khi sắp xếp

Sample Input 0

```
15
63 40
19 28
4 48
26 64
52 46
76 86
16 81
57 28
54 95
80 10
17 36
10 19
37 74
9 55
32 85
```

Sample Output 0

```
80 10
10 19
57 28
19 28
17 36
63 40
52 46
4 48
9 55
26 64
37 74
16 81
32 85
76 86
54 95
```

[Comparator_Binary Search]. Bài 10. Sắp xếp pair 2

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho mảng pair A[] gồm N pair, sắp xếp mảng pair $\text{abs}(\text{first} - \text{second})$ tăng dần, nếu cùng $\text{abs}(\text{first} - \text{second})$ thì sắp theo first tăng dần, nếu first bằng nhau thì cho second giảm dần

Input Format

- Dòng 1 là N : số lượng phần tử trong mảng
- N dòng tiếp theo là N pair trong mảng

Constraints

- $1 \leq N \leq 10^6$
- Các phần tử của pair là số int

Output Format

In ra các pair sau khi sắp xếp

Sample Input 0

```
13
12 39
13 38
0 37
25 19
2 4
47 28
21 41
24 20
43 32
10 39
12 49
4 0
12 24
```

Sample Output 0

```
2 4
4 0
24 20
25 19
43 32
12 24
47 28
21 41
13 38
12 39
10 39
0 37
12 49
```

[Comparator_Binary Search].

Bài 11. First position

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho mảng $A[]$ gồm N phần tử được sắp xếp theo thứ tự tăng dần, nhiệm vụ của bạn là tìm vị trí đầu tiên của X trong mảng hoặc kết luận là X không xuất hiện.

Lưu ý bài này các bạn cần code bằng 2 cách : Tự cài đặt theo hướng dẫn trong slide và sử dụng hàm `lower_bound`. Độ phức tạp cần đạt được cho code là $O(\log N)$

Input Format

- Dòng 1 là N và X : số lượng phần tử trong mảng và X
- Dòng 2 gồm N số viết cách nhau 1 dấu cách

Constraints

- $1 \leq N \leq 1000$
- $0 \leq A[i] \leq 10^9$

Output Format

In ra vị trí đầu tiên (theo chỉ số tính từ 0) của phần tử X hoặc in ra -1 nếu X không xuất hiện trong mảng.

Sample Input 0

```
16 9
0 1 2 3 3 3 7 7 8 8 11 11 12 12 15 15
```

Sample Output 0

```
-1
```

Sample Input 1

```
10 3
1 1 1 2 2 2 3 3 3 3
```

Sample Output 1

6

[Comparator_Binary Search]. Bài 12. Last position

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho mảng $A[]$ gồm N phần tử được sắp xếp theo thứ tự tăng dần, nhiệm vụ của bạn là tìm vị trí cuối cùng của X trong mảng hoặc kết luận là X không xuất hiện.

Lưu ý bài này các bạn cần code bằng 2 cách : Tự cài đặt theo hướng dẫn trong slide và sử dụng hàm `upper_bound`. Gợi ý là tìm `upper_bound` vs X rồi dịch trái 1 phần tử. Độ phức tạp cần đạt được cho code là $O(\log N)$

Input Format

- Dòng 1 là N và X : số lượng phần tử trong mảng và X
- Dòng 2 gồm N số viết cách nhau 1 dấu cách

Constraints

- $1 \leq N \leq 1000$
- $0 \leq A[i] \leq 10^9$

Output Format

In ra vị trí cuối cùng (theo chỉ số tính từ 0) của phần tử X hoặc in ra -1 nếu X không xuất hiện trong mảng.

Sample Input 0

```
6 3
1 2 3 3 3 4
```

Sample Output 0

4

[Comparator_Binary Search].

Bài 13. Lớn nhất nhỏ hơn hoặc bằng X

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho mảng $A[]$ gồm N phần tử được sắp xếp theo thứ tự tăng dần, nhiệm vụ của bạn là giá trị của phần tử lớn nhất nhỏ hơn hoặc bằng X trong mảng hoặc kết luận là không xuất hiện.

Lưu ý bài này các bạn cần code bằng 2 cách : Tự cài đặt theo hướng dẫn trong slide và sử dụng hàm `upper_bound`. Độ phức tạp cần đạt được cho code là $O(\log N)$

Input Format

- Dòng 1 là N và X : số lượng phần tử trong mảng và X
- Dòng 2 gồm N số viết cách nhau 1 dấu cách

Constraints

- $1 \leq N \leq 1000$
- $0 \leq A[i] \leq 10^9$

Output Format

In ra đáp án của bài toán hoặc in ra NOT FOUND nếu không tìm thấy phần tử lớn nhất $\leq X$

Sample Input 0

```
10 3
1 2 2 2 2 4 5 6 7 110
```

Sample Output 0

```
2
```

[Comparator_Binary Search].

Bài 14. Nhỏ hơn

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho mảng $A[]$, $B[]$ gồm N và M phần tử, nhiệm vụ của bạn là với mỗi phần tử trong mảng $A[]$ bạn hãy chỉ ra có bao nhiêu phần tử trong mảng $B[]$ nhỏ hơn nó.

Gợi ý : Sort mảng $B[]$ rồi dùng `binary_search` biến đổi hoặc `lower_bound`, `upper_bound`.

Input Format

Dòng 1 là N và M , dòng 2 là mảng $A[]$, dòng 3 là mảng $B[]$

Constraints

- $1 \leq N, M \leq 10^6$
- $0 \leq A[i], B[i] \leq 10^9$

Output Format

In ra kết quả của bài toán

Sample Input 0

```
5 8
2 8 7 2 8
9 3 7 1 0 4 1 5
```

Sample Output 0

```
3 7 6 3 7
```

[Comparator_Binary Search].

Bài 15. Lớn hơn

- [Problem](#)

- [Submissions](#)
- [Discussions](#)

Cho mảng $A[]$, $B[]$ gồm N và M phần tử, nhiệm vụ của bạn là với mỗi phần tử trong mảng $A[]$ bạn hãy chỉ ra có bao nhiêu phần tử trong mảng $B[]$ lớn hơn nó.

Gợi ý : Sort mảng $B[]$ rồi dùng `binary_search` biến đổi hoặc `lower_bound`, `upper_bound`.

Input Format

Dòng 1 là N và M , dòng 2 là mảng $A[]$, dòng 3 là mảng $B[]$

Constraints

- $1 \leq N, M \leq 10^6$
- $0 \leq A[i], B[i] \leq 10^9$

Output Format

In ra kết quả của bài toán

Sample Input 0

```
13 8
6 2 0 2 2 9 2 7 0 9 8 7 7
7 7 3 4 6 3 0 8
```

Sample Output 0

```
3 7 7 7 7 0 7 1 7 0 0 1 1
```

[Comparator_Binary Search]. Bài 16. Lớn hơn hoặc bằng

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho mảng $A[]$, $B[]$ gồm N và M phần tử, nhiệm vụ của bạn là với mỗi phần tử trong mảng $A[]$ bạn hãy chỉ ra có bao nhiêu phần tử trong mảng $B[]$ lớn hơn hoặc bằng nó.

Gợi ý : Sort mảng $B[]$ rồi dùng `binary_search` biến đổi hoặc `lower_bound`, `upper_bound`.

Input Format

Dòng 1 là N và M , dòng 2 là mảng $A[]$, dòng 3 là mảng $B[]$

Constraints

- $1 \leq N, M \leq 10^6$
- $0 \leq A[i], B[i] \leq 10^9$

Output Format

In ra kết quả của bài toán

Sample Input 0

```
11 9
8 1 5 9 6 7 4 0 8 6 2
8 4 0 4 8 6 2 8 5
```

Sample Output 0

```
3 8 5 0 4 3 7 9 3 4 8
```

[Comparator_Binary Search]. Bài 17. Số trong đoạn

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho mảng $A[]$ gồm N phần tử, có nhiều đoạn $[L, R]$ và bạn phải trả lời trong mảng $A[]$ có bao nhiêu phần tử thuộc đoạn $[L, R]$.

Gợi ý : Sort mảng a 1 lần, tìm vị trí đầu tiên của phần tử $\geq L$, tìm vị trí cuối cùng của phần tử $\leq R$, lấy vị trí cuối - vị trí đầu + 1 là ra kết quả. Độ phức tạp $O(N \log N)$

Input Format

- Dòng 1 là N : Số lượng phần tử trong mảng
- Dòng 2 là N số trong mảng
- Dòng 3 là Q : Số lượng đoạn [L, R]
- Q dòng tiếp theo mỗi dòng là 2 số L, R

Constraints

- $1 \leq N \leq 10^6$
- $0 \leq A[i] \leq 10^9$
- $1 \leq Q \leq 1000$
- $0 \leq L \leq R \leq 10^9$

Output Format

In ra đáp án trên từng dòng

Sample Input 0

```
9
14 14 0 8 2 0 13 2 8
5
2 10
6 6
1 5
3 4
1 3
```

Sample Output 0

```
4
0
2
0
2
```

[Comparator_Binary Search]. Bài 18. Đếm cặp số

- [Problem](#)
- [Submissions](#)

- [Discussions](#)

Cho mảng $A[]$ gồm N phần tử và số nguyên K , bạn hãy đếm xem trong mảng có bao nhiêu cặp phần tử mà có độ chênh lệch giữa chúng bằng K .

Gợi ý : Sort mảng $A[]$, duyệt qua từng chỉ số i trong mảng, với mỗi chỉ số i tìm vị trí đầu tiên và vị trí cuối cùng của $A[i] + K$ để đếm số cặp phần tử có thể kết hợp với $A[i]$ để tạo thành cặp có chênh lệch bằng K .

Input Format

- Dòng 1 gồm N và K
- Dòng 2 gồm N phần tử trong mảng $A[]$

Constraints

- $1 \leq N \leq 10^6$
- $0 \leq K \leq 10^6$
- $0 \leq A[i] \leq 10^9$

Output Format

In ra số cặp thỏa mãn

Sample Input 0

```
10 2
4 2 2 6 0 8 3 1 0 7
```

Sample Output 0

```
9
```

[Set Map]. Bài 2. Truy vấn phần tử trong mảng.

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho dãy số $A[]$ gồm có N phần tử, bạn hãy trả lời các truy vấn để xác định xem phần tử X nào đó có nằm trong mảng hay không? Bạn hãy thử giải bài này bằng 3 cách : Set, Map, Binary Search.

Nếu dùng set, map để lưu mảng $A[]$ mỗi truy vấn tìm kiếm (hàm find) bạn chỉ mất độ phức tạp $O(\log N)$ thay vì $O(N)$ như duyệt tuần tự, tương tự bạn có thể sort mảng $A[]$ để áp dụng tìm kiếm nhị phân trên mỗi truy vấn.

Input Format

- Dòng đầu tiên là số nguyên N .
- Dòng thứ 2 gồm N số nguyên $A[i]$
- Dòng thứ 3 là số lượng truy vấn Q .
- Q dòng tiếp theo mỗi dòng là một số nguyên X .

Constraints

$1 \leq N \leq 10^5$; $0 \leq A[i]$, $X \leq 10^9$; $1 \leq Q \leq 1000$;

Output Format

Đối với mỗi truy vấn in ra YES nếu phần tử X xuất hiện trong mảng, ngược lại in ra NO.

Sample Input 0

```
5
1 2 1 3 4
2
1
6
```

Sample Output 0

```
YES
NO
```