

Học chung TST2021 buổi 2

Trịnh Hữu Gia Phúc

26/02/2021

Đề bài gồm có 06 trang.

A1 - Trồng cây 1 - TREE_1

Một người làm vườn được giao nhiệm vụ trồng n cây trên một đường thẳng. Các vị trí trồng cây được đánh số từ 1 đến n từ trái sang phải. Cây trồng ở vị trí thứ i có thể được gọi là cây thứ i .

Các cây được trồng có thể gồm nhiều loại, vì thế các cây sẽ thay đổi độ cao khác nhau khi lớn lên và khi được cắt tỉa. Cụ thể, độ cao ở ngày thứ t của cây thứ i được tính bằng hàm số $h_i(t) = a_i t + b_i$, với a_i và b_i là các thông số riêng cho mỗi cây i .

Để đảm bảo hàng cây đẹp nhất có thể, người ta đôi khi sẽ thay một cây bởi một cây khác. Để đảm bảo rằng hàng cây không vi phạm quy định về khoảng cách an toàn đường dây điện của chính phủ, người ta đôi khi cần kiểm tra độ cao của cây lớn nhất trong tất cả các cây được trồng.

Các thao tác trên sẽ được cho bằng một trong 2 dạng:

- **+ t i x y**: Trong ngày thứ t , người ta sẽ thay thế cây thứ i bằng một cây mới có $a_i = x$ và $b_i = y$.
- **? t**: Hỏi rằng ở cuối ngày thứ t , độ cao của cây cao nhất trong hàng là bao nhiêu?

Người ta có thể thay đổi nhiều cây trong cùng một ngày, tuy nhiên sẽ không bao giờ thay đổi một cây nhiều lần trong cùng một ngày.

Hãy lập trình để giúp người làm vườn quản lí hàng cây.

Dữ liệu vào

- Dòng đầu tiên chứa một số nguyên n ($1 \leq n \leq 10^5$) là số lượng cây.
- Mỗi dòng trong số n dòng tiếp theo chứa hai số nguyên a_i, b_i là thông số ban đầu của các cây. ($-10^6 \leq a_i, b_i \leq 10^6$).
- Dòng tiếp theo chứa một số nguyên m ($1 \leq m \leq 10^5$) là số lượng truy vấn cần xử lí.
- Mỗi dòng trong số m dòng tiếp theo chứa các truy vấn theo dạng đã nêu. Dữ liệu đảm bảo rằng:

- $1 \leq t_j \leq t_{j+1} \leq 10^9$ với mọi $1 \leq j < m$.
- $1 \leq i_j \leq n$ với mọi $1 \leq j \leq m$.
- $-10^6 \leq x_j, y_j \leq 10^6$ với mọi $1 \leq j \leq m$.

Dữ liệu ra

- Với mỗi truy vấn **?**, in ra độ cao lớn nhất của các cây ở ngày tương ứng. Các kết quả cần được in theo thứ tự xuất hiện của truy vấn trong dữ liệu vào.

Input	Output
5	
1 3	
2 -3	
3 -5	
0 3	11
1 10	12
6	20
? 1	48
+ 2 4 10 -10	
? 2	
? 3	
? 4	
+ 4 1 9 12	

Ví dụ

Giới hạn

- 50% số test có $n, m \leq 1000$
- 50% số test còn lại có $n, m \leq 10^5$

A2 - Trồng cây 2 - TREE_2

Một người làm vườn được giao nhiệm vụ trồng n cây trên một đường thẳng. Các vị trí trồng cây được đánh số từ 1 đến n từ trái sang phải. Cây trồng ở vị trí thứ i có thể được gọi là cây thứ i . Các cây được trồng có thể gồm nhiều loại, vì thế các cây sẽ thay đổi độ cao khác nhau khi lớn lên và khi được cắt tỉa. Cụ thể, độ cao ở ngày thứ t của cây thứ i được tính bằng hàm số $h_i(t) = a_i t + b_i$, với a_i và b_i là các thông số riêng cho mỗi cây i .

Để đảm bảo hàng cây đẹp nhất có thể, người ta đôi khi sẽ thay một cây bởi một cây khác. Để đảm bảo rằng hàng cây không vi phạm quy định về khoảng cách an toàn đường dây điện của chính phủ, người ta đôi khi cần kiểm tra độ cao của cây lớn nhất trong một số cây liên tiếp.

Các thao tác trên sẽ được cho bằng một trong 2 dạng:

- **+ t i x y**: Trong ngày thứ t , người ta sẽ thay thế cây thứ i bằng một cây mới có $a_i = x$ và $b_i = y$.
- **? t l r**: Hỏi rằng ở cuối ngày thứ t , độ cao của cây cao nhất trong các cây từ l đến r là bao nhiêu?

Người ta có thể thay đổi nhiều cây trong cùng một ngày, tuy nhiên sẽ không bao giờ thay đổi một cây nhiều lần trong cùng một ngày.

Hãy lập trình để giúp người làm vườn quản lí hàng cây.

Dữ liệu vào

- Dòng đầu tiên chứa một số nguyên n ($1 \leq n \leq 10^5$) là số lượng cây.
- Mỗi dòng trong số n dòng tiếp theo chứa hai số nguyên a_i, b_i là thông số ban đầu của các cây. ($-10^6 \leq a_i, b_i \leq 10^6$).
- Dòng tiếp theo chứa một số nguyên m ($1 \leq m \leq 10^5$) là số lượng truy vấn cần xử lí.
- Mỗi dòng trong số m dòng tiếp theo chứa các truy vấn theo dạng đã nêu. Dữ liệu đảm bảo rằng:

- $1 \leq t_j \leq t_{j+1} \leq 10^9$ với mọi $1 \leq j < m$.
- $1 \leq i_j \leq n$ với mọi $1 \leq j \leq m$.
- $-10^6 \leq x_j, y_j \leq 10^6$ với mọi $1 \leq j \leq m$.
- $1 \leq l_j \leq r_j \leq n$ với mọi $1 \leq j \leq m$.

Dữ liệu ra

- Với mỗi truy vấn $?$, in ra độ cao lớn nhất của các cây ở ngày tương ứng. Các kết quả cần được in theo thứ tự xuất hiện của truy vấn trong dữ liệu vào.

Ví dụ

Input	Output
5	
1 3	
2 -3	
3 -5	
0 3	11
1 10	3
5	12
? 1 1 5	20
? 1 2 4	
+ 2 4 10 -10	
? 2 4 5	
? 3 4 5	

Giới hạn

- 50% số test có $n, m \leq 1000$
- 50% số test còn lại có $n, m \leq 10^5$

B - Phá hash - HACK_HASH

Bài này là một bài interactive, các bạn sẽ phải flush output mỗi khi in ra truy vấn. Trong C++, dùng `cout.flush()` hoặc `fflush(stdout)`. `fflush(stdout)` cùng với `ios_base::sync_with_stdio(0)` và `cout.tie(0)` sẽ không hoạt động.

Thuật toán hash có thể sử dụng để giải rất nhiều bài toán, tuy nhiên có xác suất sẽ bị sai, cụ thể là cho hai xâu khác nhau là bằng nhau. Thuật toán hash cổ điển có thể được biểu diễn bằng đoạn code sau:

```
hash=0;
for(char i: s){
    hash=(hash*base+i)%mod;
}
```

Trong đó `base` thường được chọn là một số lớn hơn số lượng số chữ cái khác nhau có thể có trong `s`, `mod` thường được chọn là một số nguyên tố lớn. Trong lập trình thi đấu, giá trị thông thường nhất của `mod` là $10^9 + 7$. Vì giá trị này rất thông dụng, người làm test thường có test để chặn các thuật toán sử dụng hash. Ngày nay, người sử dụng hash thường chọn `mod` là một số nguyên tố khác để tránh bị hack. Liệu bạn có thể hash một thuật toán hash được không?

Trong bài này, chương trình giám khảo sẽ có một hàm hash, có dạng như hàm hash ví dụ. Bạn có thể có các tương tác sau:

- In ra ? s: Hỏi mã hash của một xâu s tùy ý.
- In ra ! a b: Trả lời rằng hai xâu a và b tuy khác nhau nhưng có cùng một mã hash.

Các giới hạn của tương tác như sau:

- Các xâu s, a, b chỉ gồm các kí tự latin in thường (từ 'a' đến 'z'). Xâu a và b phải có cùng độ dài.
- Tương tác loại ? chỉ được gọi tối đa 150000 lần. Tương tác loại ! chỉ được gọi tối đa 1 lần.
- Không có giới hạn gì khác về độ dài của các xâu, tuy nhiên bạn vẫn phải tuân thủ giới hạn bộ nhớ và giới hạn thời gian.

Chương trình chấm điểm hoạt động như sau:

- Đọc vào từng dòng. Nếu dòng không bắt đầu bằng ?/! thì bạn bị đánh sai đó và nhận được thông báo "Wrong query format!" ("Truy vấn sai định dạng!").
- Nếu truy vấn không thỏa mãn giới hạn thứ nhất trong các giới hạn, bạn sẽ bị đánh sai và nhận được thông báo "Invalid query!" ("Truy vấn không hợp lệ!").
- Nếu truy vấn không thỏa mãn giới hạn thứ hai, bạn sẽ bị đánh sai và nhận được thông báo "Used too many queries!" ("Sử dụng quá nhiều truy vấn!").
- Nếu truy vấn không thỏa mãn giới hạn thứ ba thì hệ thống sẽ trả về "Memory limit exceeded" hoặc "Time limit exceeded", tương ứng với sử dụng quá nhiều bộ nhớ hoặc sử dụng quá nhiều thời gian.
- Sau khi in ra truy vấn loại !, chương trình chấm sẽ dừng hoạt động và không nhận thêm bất kì truy vấn nào. Nếu chương trình của bạn không kết thúc đúng cách, có thể dẫn đến bị lỗi TLE hoặc MLE.
- Nếu hai xâu a và b có hash khác nhau, bạn sẽ bị đánh sai và nhận được thông báo "Wrong answer!" ("Sai kết quả!").
- Nếu hai xâu a và b giống hệt nhau, bạn sẽ bị đánh sai và nhận được thông báo "Identical strings!" ("Xâu giống nhau!")
- Nếu hai xâu a và b khác nhau, có cùng độ dài, nhưng có cùng mã hash thì bạn được cho là đúng và nhận được thông báo "Correct answer!" ("Kết quả đúng!").

Ví dụ

Input	Output
15	? a
16	? b
5634745245	? ababababab
5634745245	? bababababa
5634745245	! ababababab bababababa

Giới hạn

- Trong 25% số test, chương trình chấm có $mod = 31$
- Trong 25% số test tiếp theo, chương trình chấm có $base = 1000000007$
- Trong 50% số test còn lại, chương trình chấm có $31 \leq base < mod \leq 10^9 + 7$

C1 - Đường tròn nhỏ nhất 1 - CIRCLE_1

Trên một mặt phẳng tọa độ cho n điểm, đánh số từ 1 đến n không theo một thứ tự nhất định. Bạn cần tìm một đường tròn nhỏ nhất có tâm ở một trong n điểm cho trước mà chứa ít nhất k điểm trong số n điểm đã cho. In ra chỉ số của điểm được chọn làm tâm.

Đây là một bài toán tối ưu, có nghĩa là bạn sẽ được chấm điểm dựa theo độ tối ưu của kết quả của bạn, không nhất thiết phải đưa ra kết quả tối ưu nhất

Cách chấm điểm như sau: Giả sử điểm bạn chọn tạo ra bình phương bán kính là x , bình phương bán kính trong kết quả của giám khảo là y thì số điểm bạn đạt được trong test đó là $\min(100, \lceil \frac{100y}{x} \rceil)$ trên thang 100. Nếu bạn tìm ra kết quả tốt hơn giám khảo, bạn cũng sẽ nhận được thêm thông báo "Better than jury!" ("Tốt hơn giám khảo!").

Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên n và k ($1 \leq k \leq n \leq 50000$).
- Dòng thứ i trong số m dòng tiếp theo chứa 2 số nguyên x_i, y_i ($-10^6 \leq x_i, y_i \leq 10^6$) là tọa độ của các điểm được cho. Dữ liệu đảm bảo không có 2 điểm nào trùng nhau.

Dữ liệu ra

- In ra trên một dòng duy nhất chỉ số của điểm tâm mà bạn cho rằng sẽ có kết quả bé nhất.

Ví dụ

Input	Output	Input	Output
8 2	3	8 8	4
4 6		4 6	
7 6		7 6	
3 4		3 4	
4 4		4 4	
10 4		10 4	
1 2		1 2	
4 2		4 2	
7 2		7 2	

Giới hạn

- Trong 40% số test có $n \leq 1000$.
- Trong 40% số test khác có $k = 2$.
- Trong 20% số test còn lại không có giới hạn gì thêm.

C2 - Đường tròn nhỏ nhất 2 - CIRCLE_2

Trên một mặt phẳng tọa độ cho n điểm, đánh số từ 1 đến n không theo một thứ tự nhất định. Bạn cần tìm một đường tròn nhỏ nhất mà chứa ít nhất k điểm trong số n điểm đã cho. In ra tọa độ của điểm được chọn làm tâm, điểm này không nhất thiết phải là một trong các điểm cho trước.

Đây là một bài toán tối ưu, có nghĩa là bạn sẽ được chấm điểm dựa theo độ tối ưu của kết quả của bạn, không nhất thiết phải đưa ra kết quả tối ưu nhất.

Cách chấm điểm như sau: Giả sử điểm bạn chọn tạo ra bình phương bán kính là x , bình phương bán kính trong kết quả của giám khảo là y thì số điểm bạn đạt được trong test đó là $\min(100, \lceil \frac{100y}{x} \rceil)$ trên thang 100. Nếu bạn tìm ra kết quả tốt hơn giám khảo, bạn cũng sẽ nhận được thêm thông báo "Better than jury!" ("Tốt hơn giám khảo!").

Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên n và k ($1 \leq k \leq n \leq 100000$).
- Dòng thứ i trong số m dòng tiếp theo chứa 2 số nguyên x_i, y_i ($-10^6 \leq x_i, y_i \leq 10^6$) là tọa độ của các điểm được cho. Dữ liệu đảm bảo không có 2 điểm nào trùng nhau.

Dữ liệu ra

- In ra trên một dòng duy nhất tọa độ của điểm tâm mà bạn cho rằng sẽ có kết quả bé nhất. Hãy in ra kết quả bằng số thực.

Ví dụ

Input	Output	Input	Output
8 2	3.5 4	8 8	5.5 3
4 6		4 6	
7 6		7 6	
3 4		3 4	
4 4		4 4	
10 4		10 4	
1 2		1 2	
4 2		4 2	
7 2		7 2	

Giới hạn

- Trong 20% số test có $k = 2$.
- Trong 30% số test khác có $k = n$.
- Trong 30% số test khác có $n \leq 50$.
- Trong 20% số test còn lại có $n \leq 200$.