



# Dress'em in Vests!

Link submit: <http://codeforces.com/problemset/problem/161/A>

Solution:

|        |   |
|--------|---|
| C++    | <a href="http://ideone.com/5Gs8C8">http://ideone.com/5Gs8C8</a>   |
| Java   | <a href="https://ideone.com/fkeGue">https://ideone.com/fkeGue</a> |
| Python | <a href="http://ideone.com/FoG4Xi">http://ideone.com/FoG4Xi</a>   |

Tóm tắt đề:

Một đội quân có  $N$  người lính, người lính thứ  $i$  có kích thước là  $a_i$ , người lính đó có thể mặc bộ vest có kích thước nằm trong đoạn  $[a_i - x, a_i + y]$ . Hiện có sẵn  $M$  bộ áo vest, áo thứ  $j$  có kích thước là  $b_j$ . Yêu cầu hãy phân chia các bộ vest sao cho số lượng người lính có vest là lớn nhất có thể và xuất ra một cách phân chia.

Input:

Dòng đầu chứa 4 số nguyên dương  $N, M, x, y$  ( $1 \leq N, M \leq 10^5, 0 \leq x, y \leq 10^9$ ).

Dòng thứ hai chứa  $N$  số nguyên dương là kích thước của những người lính ( $1 \leq a_i \leq 10^9$ ) được liệt kê theo thứ tự không giảm.

Dòng thứ ba chứa  $M$  số nguyên dương là kích thước của những bộ áo vest ( $1 \leq b_j \leq 10^9$ ) được liệt kê theo thứ tự không giảm.

Output:

Dòng đầu tiên chứa số nguyên  $K$  là cách phân chia lớn nhất có thể tạo được.

$K$  dòng tiếp theo, mỗi dòng chứa 1 cặp  $u_i, v_i$  tức người lính thứ  $u_i$  sẽ mang bộ vest  $v_i$  ( $1 \leq u_i \leq N, 1 \leq v_i \leq M$ ).

Nếu có nhiều cách thì xuất ra một cách bất kì.

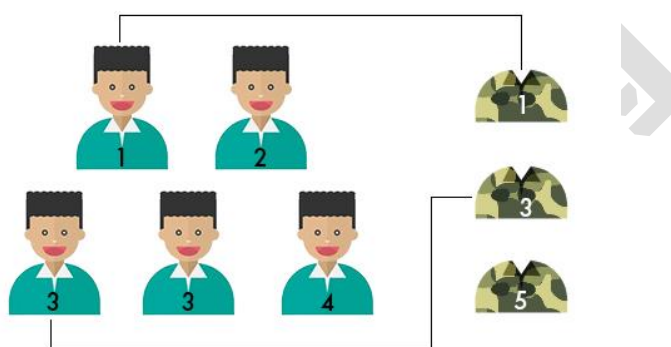
Ví dụ:

|           |     |
|-----------|-----|
| 5 3 0 0   | 2   |
| 1 2 3 3 4 | 1 1 |
| 1 3 5     | 3 2 |

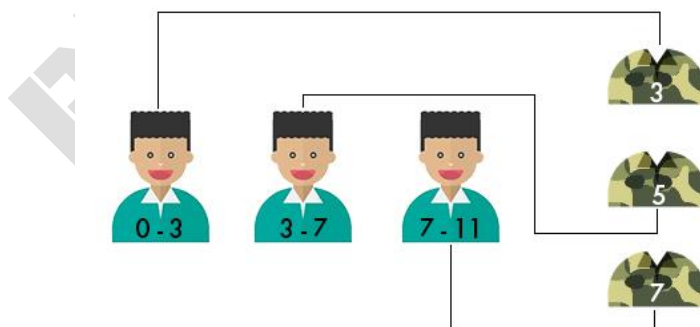
|         |     |
|---------|-----|
| 3 3 2 2 | 3   |
| 1 5 9   | 1 1 |
| 3 5 7   | 2 2 |
|         | 3 3 |

### Giải thích ví dụ:

**Ví dụ 1:** Vì  $x$  và  $y$  bằng 0 nên mỗi người chỉ có thể mang bộ vest có kích thước đúng bằng kích thước của họ. Nên người 1 có thể mang được bộ vest 1, người thứ 2 và thứ 5 không thể mang được bộ nào. Bộ vest thứ 2 có thể được mang bởi người thứ 3 hoặc 4. Vì vậy số người tối đa có thể mang là 2 và 1 trong những cách chia đó là (1-1, 3-2), hoặc cách khác là (1-1, 4-2).



**Ví dụ 2:** Người 1 có thể mang vest kích thước trong đoạn  $[1-2; 1+2]$  tức đoạn  $[-1, 3]$  nên mang được vest thứ nhất, tương tự 2 người còn lại cũng lần lượt mang được bộ vest 2 và 3 nên kết quả là 3.



### Hướng dẫn giải:

Trước tiên, ta thấy rằng trong trường hợp một người có thể mang nhiều bộ vest và một bộ vest có thể được mang bởi nhiều người, thì ta nên cho người nhỏ nhất trong nhóm đó mang bộ vest nhỏ nhất có thể, bởi vì nếu chia như vậy, thì những bộ vest lớn hơn có thể để lại cho những người lính khác có kích thước lớn hơn có thể mang được.

Tức ý tưởng trước tiên ta sẽ sắp xếp lại 2 mảng tăng dần.

Sau đó, theo cách đơn giản nhất, với mỗi người lính  $i$ , ta sẽ tìm bộ vest  $j$  nhỏ nhất chưa có người mang và có kích thước nằm trong đoạn  $[a_i - x, a_i + y]$ . Tức ta sẽ có thuật toán như sau:

```
for each soldier i in army:
    for each vest j in list vests:
        if vest j is not used and in range [a[i] - x, a[i] + y]:
            push (i, j) to result
            break
```

Tuy nhiên, với cách này, ta tốn độ phức tạp là  $O(N*M)$ , với  $N, M$  có giới hạn  $10^5$  là không khả thi. Ta cần phải cải tiến thuật toán, thì ở đây, ta thấy rằng nếu người lính  $i$  được chọn mang bộ vest  $j$ , thì người lính  $i+1$  trở về sau, chắc chắn chỉ có thể mang các bộ vest từ  $j+1$  trở về sau. Bởi nếu các bộ vest từ  $j$  trở về trước mà có thể dùng được, thì người lính  $i$  đã dùng rồi (vì ta ưu tiên chọn người lính nhỏ mang vest nhỏ). Như vậy nghĩa là các bộ vest từ 1 đến  $j$  hoặc đã được dùng, hoặc không thể dùng, nên ta không cần kiểm tra lại.

Tức ta có phương pháp cải tiến như sau:

```
j = 0
for i from 1 to n:
    while j <= m
        if b[j] in range [a[i] - x, a[i] + y]:
            push (i, j) to result
            break
        j += 1
```

Độ phức tạp:  $O(\max(N, M))$  với  $N$  và  $M$  lần lượt là số lượng người lính và số lượng bộ vest.