

## Bài 1: TWOELE

File dữ liệu vào: `stdin`  
File kết quả: `stdout`  
Hạn chế thời gian: `1s`  
Hạn chế bộ nhớ: `512M`

Cho dãy số nguyên  $a_i$  có  $n$  phần tử đánh số từ 1 đến  $n$ . Tìm  $i$  và  $j$  sao cho  $i < j$  và  $(a_i + a_j) \times (j - i)$  là lớn nhất.

### Dữ liệu

- Dòng đầu tiên chứa số nguyên  $n$  ( $2 \leq n \leq 10^6$ ).
- Dòng thứ hai chứa  $n$  số nguyên  $a_i$  ( $1 \leq a_i \leq 10^6$ ).

### Kết quả

Đưa ra kết quả lớn nhất tìm được.

### Ví dụ

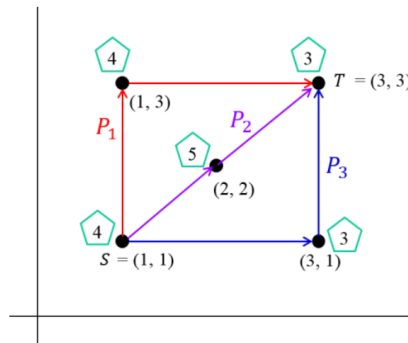
<code>stdin</code>	<code>stdout</code>
5 1 3 2 5 4	21
5 8 3 6 3 1	36

## Bài 2: ELECAR

File dữ liệu vào:	<code>stdin</code>
File kết quả:	<code>stdout</code>
Hạn chế thời gian:	2s
Hạn chế bộ nhớ:	512M

Một xe chạy bằng điện EV mới được sử dụng thay thế cho các xe nhiên liệu truyền thống. Có  $n$  thành phố trên một mặt phẳng, với hai thành phố bất kỳ sẽ có một con đường nối chúng. Giả sử hai thành phố  $A$  và  $B$  có tọa độ là  $(a, b)$  và  $(c, d)$  thì khoảng cách giữa hai thành phố này  $d(A, B) = |a - c| + |b - d|$ . Để dễ hình dung, ta coi mỗi đơn vị di chuyển EV sẽ tiêu tốn đúng 1 đơn vị nhiên liệu điện. Mỗi thành phố có một trạm nạp điện cho xe, thành phố  $A$  sẽ có chi phí  $c(A)$  cho một đơn vị nhiên liệu điện. Nếu xe EV tại thành phố  $A$  đang không có đơn vị nhiên liệu nào muốn di chuyển đến thành phố  $B$  thì nó phải mất chi phí nhỏ nhất là  $c(A) \times d(A, B)$  để có thể di chuyển từ  $A$  đến  $B$ .

ATM phải di chuyển từ thành phố  $S$  đến thành phố  $T$  bằng xe EV của anh ấy. Ban đầu chiếc xe này đang không có nhiên liệu nào. Và chiếc xe này có thể nạp đầy được  $W$  nhiên liệu hay chiếc EV này không thể chứa nhiều hơn  $W$  nhiên liệu một lúc. Do đó nó không thể di chuyển được giữa hai thành phố có khoảng cách lớn hơn  $W$ . Ngoài ra ATM chỉ cho phép nạp tối đa  $p$  lần nạp điện kể cả lần nạp đầu tiên.



Cho biết tọa độ của  $n$  thành phố và  $S, T$ , chi phí nạp nhiên liệu tại mỗi thành phố,  $W$  và  $p$ , hãy tính chi phí nhỏ nhất để có thể đi được từ  $S$  đến  $T$ .

### Dữ liệu

- Dòng đầu tiên chứa số nguyên  $n$  ( $1 < n \leq 1000$ ).
- $n$  dòng tiếp theo mỗi dòng chứa 3 số nguyên  $a, b, c$  là tọa độ và chi phí của các thành phố ( $0 \leq a, b \leq 10^6, 1 \leq c \leq 10^4$ ). Trong đó dòng đầu tiên là thành phố  $S$ , dòng thứ hai là thành phố  $T$ .
- Tiếp theo chứa số nguyên  $W$  ( $1 \leq W \leq 10^5$ ).
- Dòng cuối cùng chứa số nguyên  $p$  ( $1 \leq p \leq 10$ ).

### Kết quả

Ghi ra chi phí nhỏ nhất, hoặc ghi ra -1 nếu không thể đi được từ  $S$  đến  $T$ .

## Ví dụ

stdin	stdout
4 0 0 1 3 0 3 1 0 3 2 0 3 4 2	3
5 1 1 4 3 3 3 1 3 4 2 2 5 3 1 3 3 2	14

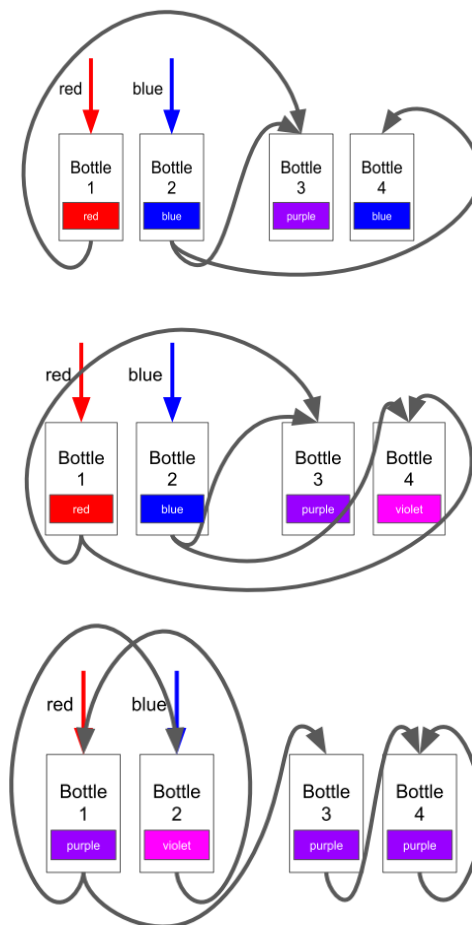
## Bài 3: INKMIX

File dữ liệu vào: `stdin`  
 File kết quả: `stdout`  
 Hạn chế thời gian: 2s  
 Hạn chế bộ nhớ: 512M

Có  $n$  bình mực, đánh số từ 1 đến  $n$ . Ban đầu tất cả các bình đều rỗng. Sau đó  $m$  bình đầu tiên được đổ mực vào, các bình này được đổ các màu mực khác nhau. Và  $n$  bình có các ống nối với nhau. Mực của các bình có thể trao đổi liên tục với nhau. Nhưng đến cuối ngày, màu mực tại mỗi bình sẽ đến trạng thái cân bằng, khi đó ta có:

- Một bình A chứa mực nếu và chỉ nếu mực ở  $m$  bình đầu tiên có thể chảy đến A thông qua một số ống nối. Ngược lại A là rỗng.
- Mỗi bình không rỗng chứa mực là một màu.
- Nếu bình B được trộn bởi nhiều màu mực khác nhau thì B gọi là bình lẫn màu.
- Một bình C không rỗng không lẫn màu sẽ chứa màu mực của bình nguồn chảy đến C.
- Một bình lẫn màu sẽ có màu khác với các bình lẫn màu khác.

Xem các ví dụ dưới đây để hiểu rõ hơn.



Đếm số màu khác nhau tại trạng thái cân bằng.

### Dữ liệu

- Dòng đầu tiên chứa số nguyên  $n, m, k$  ( $1 \leq m \leq n \leq 10^5, 1 \leq k \leq 5 \times 10^5$ ).
- $k$  dòng tiếp theo mỗi dòng chứa 2 số nguyên  $a, b$  mô tả ống nối sẽ chảy từ bình  $a$  sang bình  $b$ .

## Kết quả

Ghi ra số màu khác nhau tại trạng thái cân bằng.

## Ví dụ

stdin	stdout
4 2 3 1 3 2 3 2 4	3
4 2 4 1 3 1 4 2 3 2 4	4
4 2 5 1 2 2 1 1 3 3 4 4 4	2
4 2 1 3 4	2