

팀 만들기 (teambuilding)

20XX년 국제정보올림피아드에서는 혼성 종목이 신설되었다. 혼성 종목에서는 남학생 한 명과 여학생 한 명이 팀을 이루어 문제를 풀게 된다. 정보올림피아드위원회에서는 혼성 종목에 참가할 대표팀을 만들기 위해 N 명의 남학생 후보와 M 명의 여학생 후보를 선발했다. N 명의 남학생 후보들에게는 0번부터 $N - 1$ 번까지의 번호가 붙어 있다. 마찬가지로, M 명의 여학생 후보들에게는 0번부터 $M - 1$ 번까지의 번호가 붙어 있다.

팀의 실력에는 팀을 구성하는 두 학생의 발상 능력과 구현 능력이 모두 영향을 준다. 학생들의 발상 능력과 구현 능력은 1 이상 10^9 이하의 정수로 나타낼 수 있다. i 번 남학생의 발상 능력은 $A_1[i]$, 구현 능력은 $B_1[i]$ 이다. 또한, j 번 여학생의 발상 능력은 $A_2[j]$, 구현 능력은 $B_2[j]$ 이다. i 번 남학생과 j 번 여학생이 한 팀을 이룬다면, 이 팀의 실력은 $(A_1[i] + A_2[j]) \times (B_1[i] + B_2[j])$ 로 정의된다.

학생들은 모두 치열한 경쟁을 거쳐 선발된 후보이기 때문에, 어떤 남학생이 다른 남학생보다 발상 능력과 구현 능력이 모두 뛰어난 경우는 없다. 더 정확하게는, 남학생들의 발상 능력은 **번호가 커짐에 따라 증가**하고, 구현 능력은 **번호가 커짐에 따라 감소**한다. 즉, $0 \leq i \leq N - 2$ 에 대해 $A_1[i] < A_1[i + 1]$ 과 $B_1[i] > B_1[i + 1]$ 이 성립한다. 마찬가지로, 여학생들 또한 발상 능력은 **번호가 커짐에 따라 증가**하고, 구현 능력은 **번호가 커짐에 따라 감소**한다. 즉, $0 \leq j \leq M - 2$ 에 대해 $A_2[j] < A_2[j + 1]$ 과 $B_2[j] > B_2[j + 1]$ 이 성립한다.

정보올림피아드위원회에서는 다양한 시나리오에 따라 팀의 실력이 최대가 되도록 팀을 구성하고자 한다. 0부터 $Q - 1$ 번까지 번호가 붙은 Q 개의 시나리오가 존재한다. k 번 시나리오에서, 남학생의 번호는 $L_1[k]$ 이상 $R_1[k]$ 이하가 되어야 하고, 여학생의 번호는 $L_2[k]$ 이상 $R_2[k]$ 이하가 되어야 한다. 각 시나리오에 대해, 조건을 만족하며 구성할 수 있는 팀의 실력의 최댓값을 구하는 프로그램을 작성하여야.

함수 목록 및 정의

여러분은 아래 함수를 구현해야 한다.

```
vector<long long> build_teams(vector<int> A1, vector<int> B1, vector<int> A2,
vector<int> B2, vector<int> L1, vector<int> R1, vector<int> L2, vector<int> R2)
```

- A_1, B_1 : 길이가 N 인 배열. i 번 남학생의 발상 능력은 $A_1[i]$, 구현 능력은 $B_1[i]$ 이다. ($0 \leq i \leq N - 1$)
- A_2, B_2 : 길이가 M 인 배열. j 번 여학생의 발상 능력은 $A_2[j]$, 구현 능력은 $B_2[j]$ 이다. ($0 \leq j \leq M - 1$)
- L_1, R_1, L_2, R_2 : 길이가 Q 인 배열. k 번 시나리오에서, 남학생의 번호는 $L_1[k]$ 이상 $R_1[k]$ 이하가 되어야 하고, 여학생의 번호는 $L_2[k]$ 이상 $R_2[k]$ 이하가 되어야 한다. ($0 \leq k \leq Q - 1$)
- 이 함수는 크기가 Q 인 배열 C 를 반환해야 한다. $C[k]$ 는 k 번 시나리오에서 구성할 수 있는 팀의 실력의 최댓값과 같아야 한다.

제출하는 소스 코드의 어느 부분에서도 입출력 함수를 실행해서는 안 된다.

제약 조건

- $1 \leq N \leq 100\,000$
- $1 \leq M \leq 100\,000$
- 모든 $0 \leq i \leq N - 1$ 에 대해 $1 \leq A_1[i], B_1[i] \leq 10^9$
- 모든 $0 \leq j \leq M - 1$ 에 대해 $1 \leq A_2[j], B_2[j] \leq 10^9$
- 모든 $0 \leq i \leq N - 2$ 에 대해 $A_1[i] < A_1[i + 1], B_1[i] > B_1[i + 1]$
- 모든 $0 \leq j \leq M - 2$ 에 대해 $A_2[j] < A_2[j + 1], B_2[j] > B_2[j + 1]$
- $1 \leq Q \leq 100\,000$
- 모든 $0 \leq k \leq Q - 1$ 에 대해 $0 \leq L_1[k] \leq R_1[k] \leq N - 1$
- 모든 $0 \leq k \leq Q - 1$ 에 대해 $0 \leq L_2[k] \leq R_2[k] \leq M - 1$

부분문제

1. (5점)

- $N \leq 500, M \leq 500, Q \leq 500.$

2. (10점)

- $Q \leq 20.$

3. (10점)

- 모든 $0 \leq k \leq Q - 1$ 에 대해 $L_2[k] = 0, R_2[k] = M - 1.$

4. (35점)

- 모든 $0 \leq k \leq Q - 1$ 에 대해 $L_2[k] = R_2[k],$

5. (40점)

- 추가적인 제약 조건이 없다.

예제 1

$N = 5, M = 4, A_1 = [2, 7, 8, 9, 10], B_1 = [10, 9, 8, 6, 1], A_2 = [1, 3, 5, 9], B_2 = [10, 8, 7, 5], Q = 3,$
 $L_1 = [0, 2, 1], R_1 = [4, 3, 1], L_2 = [1, 0, 0], R_2 = [3, 2, 0]$ 인 경우를 생각해 보자.

그레이더는 다음과 같이 함수를 호출한다.

```
build_teams([2, 7, 8, 9, 10], [10, 9, 8, 6, 1], [1, 3, 5, 9], [10, 8, 7, 5],
[0, 2, 1], [4, 3, 1], [1, 0, 0], [3, 2, 0])
```

0번 시나리오에서, 남학생의 번호는 0 이상 4 이하, 여학생의 번호는 1 이상 3 이하가 되어야 한다. 1번 남학생과 3번 여학생으로 팀을 구성하면 팀의 실력은 $(7+9) \times (9+5) = 224$ 가 된다. 이것은 조건을 만족하며 구성할 수 있는 팀의 실력 중 최댓값이다.

1번 시나리오에서, 남학생의 번호는 2 이상 3 이하, 여학생의 번호는 0 이상 2 이하가 되어야 한다. 2번 남학생과 2번 여학생으로 팀을 구성하면 팀의 실력은 $(8+5) \times (8+7) = 195$ 가 된다. 이것은 조건을 만족하며 구성할 수 있는 팀의 실력 중 최댓값이다.

2번 시나리오에서, 남학생의 번호는 1이고 여학생의 번호는 0이어야 한다. 팀의 실력은 $(7+1) \times (9+10) = 152$ 이다.

따라서 함수는 $[224, 195, 152]$ 를 반환해야 한다.

예제 2

$N = 4, M = 2, A_1 = [1, 6, 8, 10], B_1 = [9, 5, 3, 1], A_2 = [5, 6], B_2 = [8, 7], Q = 4, L_1 = [0, 1, 2, 3], R_1 = [0, 1, 2, 3], L_2 = [0, 0, 0, 0], R_2 = [1, 1, 1, 1]$ 인 경우를 생각해 보자.

그레이더는 다음과 같이 함수를 호출한다.

```
build_teams([1, 6, 8, 10], [9, 5, 3, 1], [5, 6], [8, 7], [0, 1, 2, 3], [0, 1, 2, 3], [0, 0, 0, 0], [1, 1, 1, 1])
```

0번 시나리오에서, 0번 남학생과 1번 여학생을 고르면 팀의 실력은 $(1+6) \times (9+7) = 112$ 가 된다.

1번 시나리오에서, 1번 남학생과 1번 여학생을 고르면 팀의 실력은 $(6+6) \times (5+7) = 144$ 가 된다.

2번 시나리오에서, 2번 남학생과 0번 여학생을 고르면 팀의 실력은 $(8+5) \times (3+8) = 143$ 가 된다.

3번 시나리오에서, 3번 남학생과 0번 여학생을 고르면 팀의 실력은 $(10+5) \times (1+8) = 135$ 가 된다.

이들은 모두 각 시나리오의 조건을 만족하며 구성할 수 있는 팀의 실력 중 최댓값이다. 따라서 함수는 $[112, 144, 143, 135]$ 를 반환해야 한다.

Sample grader

Sample grader는 아래와 같은 형식으로 입력을 받는다.

- Line 1: $N M$
- Line $2 + i$ ($0 \leq i \leq N - 1$): $A_1[i] B_1[i]$
- Line $2 + N + j$ ($0 \leq j \leq M - 1$): $A_2[j] B_2[j]$
- Line $2 + N + M$: Q
- Line $3 + N + M + k$ ($0 \leq k \leq Q - 1$): $L_1[k] R_1[k] L_2[k] R_2[k]$

Sample grader는 다음을 출력한다.

- Line $1 + k$ ($0 \leq k \leq Q - 1$): 함수 `build_teams`가 반환한 배열의 k 번째 원소

Sample grader는 실제 채점에서 사용하는 그레이더와 다를 수 있음에 유의하라.