

회의실 2 (meeting2)

기업 KDH에서는 N 개의 회의를 매일 진행한다. 회의에는 0부터 $N - 1$ 까지의 번호가 붙어져 있으며 모든 $0 \leq i \leq N - 1$ 에 대해 i 번 회의는 시각 $S[i]$ 에 시작해 시각 $E[i]$ 에 끝난다.

KDH에서 회의를 여는 방식은 특별하다. 어떤 날에 진행되는 서로 다른 회의 i 와 j 가 다음 조건 중 적어도 하나를 만족하면 두 회의는 해당 날에 서로 **관련있는 회의**라고 부른다:

- 두 회의가 동시에 진행되는 시각이 존재한다.
- 두 회의와 동시에 관련있는 해당 날에 진행되는 회의 k 가 존재한다.

두 회의 i 와 j 가 서로 관련있는 회의가 아니라면 두 회의는 해당 날에 서로 **관련없는 회의**라고 부른다.

KDH는 매일 회의를 진행할 때 각 회의를 특정 회의실에 배정하여 진행한다. 이 때, 해당 날에 서로 관련없는 회의가 같은 회의실에 배정되지 **않아야** 한다. KDH에서는 이러한 조건을 만족하는 배정 방법들 중 필요한 회의실의 수가 최소인 방법을 선택할 것이다. 이러한 배정에서 필요한 최소 회의실의 수를 회의들의 **비용**이라고 하자.

KDH는 현재 회의들에 불필요하게 많은 자원이 소모된다고 판단해 회의의 수를 단 하나로 줄이기로 결정하였다. 이를 위해, KDH는 $N - 1$ 일에 걸쳐 매일 다음과 같은 작업을 반복한다:

- 아직 취소하지 않은 하나의 회의를 선택한다.
- 그 날부터 선택된 회의를 영구적으로 취소한다.
- 취소되지 않은 회의들을 전부 진행한다.

이 과정이 모두 끝나게 되면 단 하나의 회의를 제외하고 모든 회의가 취소된다. 마지막에 남는 회의가 무엇인지는 상관 없다.

KDH는 더욱 비용을 절감하기 위해 여러 방법들 중 $N - 1$ 일 동안 각 날에 필요한 **비용의 합**이 최소인 방법을 선택하려고 한다. 당신은 KDH를 위해 이러한 방법이 얼마나 존재하는지 구해야 한다. 두 방법이 같다는 것은, 각 날에 **취소하기로 선택한 회의들이 모두 같다는 것**을 뜻한다: 구체적으로, $N - 1$ 일에 걸쳐 취소하기로 선택한 회의가 모두 같을 경우, 회의실에 남은 회의들을 배정하는 방법이 다르더라도 이는 같은 경우로 간주한다. 단, 방법의 수가 매우 커질 수 있으므로 소수 1000 000 007로 나눈 나머지를 구해야 한다.

함수 목록 및 정의

여러분은 아래 함수를 구현해야 한다.

```
int count_removals(vector<int> S, vector<int> E)
```

- S, E : 크기가 N 인 정수 배열. 모든 $0 \leq i \leq N - 1$ 에 대해, i 번 회의는 시각 $S[i]$ 에 시작해 시각 $E[i]$ 에 끝난다.
- 이 함수는 $N - 1$ 일 동안 각 날에 필요한 비용의 합을 최소화하도록 각 날에 취소할 회의를 정하는 방법의 수를 1000 000 007로 나눈 나머지를 반환해야 한다.

제출하는 소스 코드의 어느 부분에서도 입출력 함수를 실행해서는 안 된다.

제약 조건

- $2 \leq N \leq 2000$
- 모든 $0 \leq i \leq N - 1$ 에 대해 $1 \leq S[i] < E[i] \leq 2N$
- 모든 $0 \leq i < j \leq N - 1$ 에 대해 $S[i] \neq S[j], S[i] \neq E[j], E[i] \neq S[j], E[i] \neq E[j]$

부분문제

1. (3점)

- $N \leq 10$
- $S[0] = 1, E[0] = 2N$

2. (8점)

- $N \leq 20$

3. (30점)

- $N \leq 300$

4. (12점)

- 한 시각에 진행되는 회의는 최대 2개이다.

5. (12점)

- 모든 $0 \leq i, j \leq N - 1$ 에 대해 $i \neq j, S[i] < S[j] < E[i] < E[j]$ 를 만족하는 i, j 쌍이 존재하지 않는다.

6. (10점)

- 모든 $0 \leq i, j \leq N - 1$ 에 대해 $i \neq j, S[i] < S[j] < E[j] < E[i]$ 를 만족하는 i, j 쌍이 존재하지 않는다.

7. (25점)

- 추가적인 제약 조건이 없다.

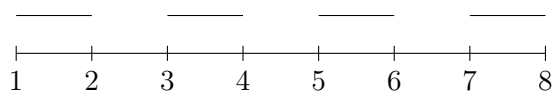
예제 1

$N = 4, S = [1, 3, 5, 7], E = [2, 4, 6, 8]$ 인 경우를 생각해 보자.

그레이더는 다음과 같이 함수를 호출한다.

```
count_removals([1, 3, 5, 7], [2, 4, 6, 8])
```

아래 그림은 각 회의가 진행되는 시간을 나타낸다.



어떠한 방법으로 진행하지 않을 회의를 정하더라도 각 날에 필요한 회의실의 수, 즉 비용이 순서대로 3, 2, 1로 같고, 모든 날에 필요한 비용의 합도 6으로 같다. 따라서 모든 방법이 가능한 방법이다.

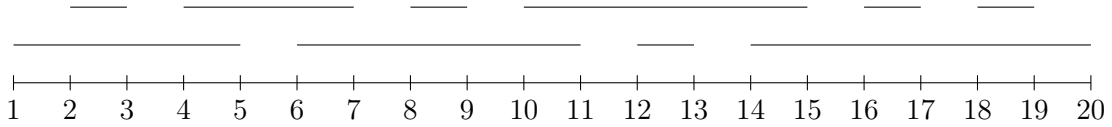
함수는 24를 반환해야 한다.

예제 2

$N = 10$, $S = [1, 2, 4, 6, 8, 10, 12, 14, 16, 18]$, $E = [5, 3, 7, 11, 9, 15, 13, 20, 17, 19]$ 인 경우를 생각해 보자.
 그레이더는 다음과 같이 함수를 호출한다.

```
count_removals([1, 2, 4, 6, 8, 10, 12, 14, 16, 18], [5, 3, 7, 11, 9, 15, 13, 20, 17, 19])
```

아래 그림은 각 회의가 진행되는 시간을 나타낸다.



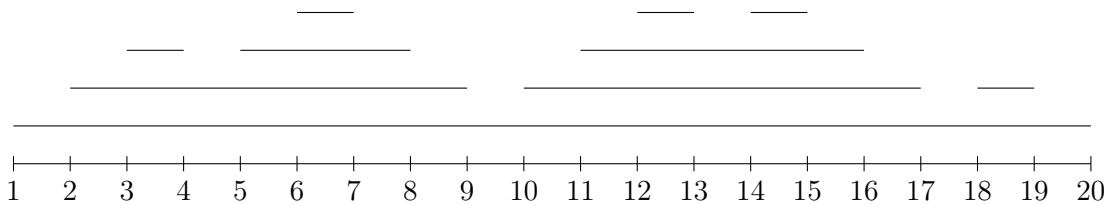
함수는 13280을 반환해야 한다.

예제 3

$N = 10$, $S = [1, 2, 3, 5, 6, 10, 11, 12, 14, 18]$, $E = [20, 9, 4, 8, 7, 17, 16, 13, 15, 19]$ 인 경우를 생각해 보자.
 그레이더는 다음과 같이 함수를 호출한다.

```
count_removals([1, 2, 3, 5, 6, 10, 11, 12, 14, 18], [20, 9, 4, 8, 7, 17, 16, 13, 15, 19])
```

아래 그림은 각 회의가 진행되는 시간을 나타낸다.



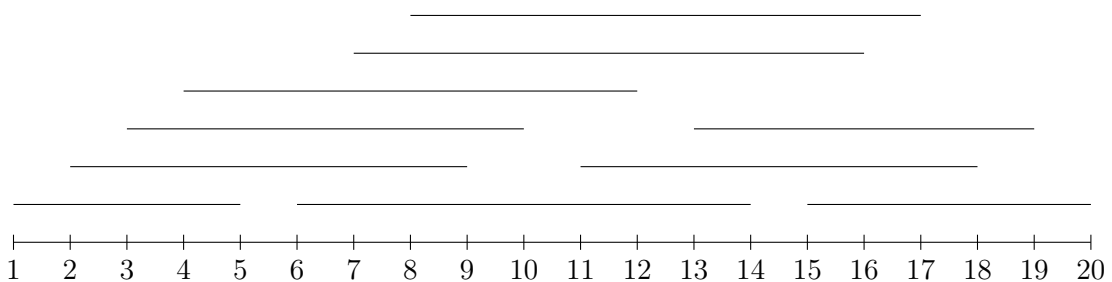
함수는 845040을 반환해야 한다.

예제 4

$N = 10$, $S = [1, 2, 3, 4, 6, 7, 8, 11, 13, 15]$, $E = [5, 9, 10, 12, 14, 16, 17, 18, 19, 20]$ 인 경우를 생각해 보자.
 그레이더는 다음과 같이 함수를 호출한다.

```
count_removals([1, 2, 3, 4, 6, 7, 8, 11, 13, 15], [5, 9, 10, 12, 14, 16, 17, 18, 19, 20])
```

아래 그림은 각 회의가 진행되는 시간을 나타낸다.



함수는 1797408을 반환해야 한다.

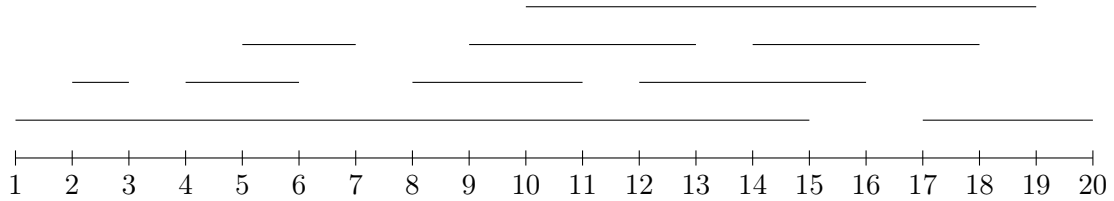
예제 5

$N = 10$, $S = [12, 5, 10, 2, 4, 17, 8, 1, 14, 9]$, $E = [16, 7, 19, 3, 6, 20, 11, 15, 18, 13]$ 인 경우를 생각해 보자.

그레이더는 다음과 같이 함수를 호출한다.

```
count_removals([12, 5, 10, 2, 4, 17, 8, 1, 14, 9], [16, 7, 19, 3, 6, 20, 11, 15, 18, 13])
```

아래 그림은 각 회의가 진행되는 시간을 나타낸다.



함수는 647760을 반환해야 한다.

Sample grader

Sample grader는 아래와 같은 형식으로 입력을 받는다.

- Line 1: N
- Line $2 + i$ ($0 \leq i \leq N - 1$): $S[i] E[i]$

Sample grader는 다음을 출력한다.

- Line 1: 함수 `count_removals`가 반환한 값

Sample grader는 실제 채점에서 사용하는 그레이더와 다를 수 있음에 유의하라.