```
def maxMeetings(meetings):
    meetings.sort(key=lambda x: x[1])

    count = 1
    last_end = meetings[0][1]

    for i in range(1, len(meetings)):
        if meetings[i][0] >= last_end:
            count += 1
            last_end = meetings[i][1]

    return count


T = int(input())
for _ in range(T):
    N = int(input())
    meetings = []
    for _ in range(N):
        s, e = map(int, input().split())
        meetings.append((s, e))

    print(maxMeetings(meetings))
```

```
1
3
1 3
2 4
3 5
2
```

```
import sys
sys.setrecursionlimit(10**7)

def max_crossing_sum(arr, left, mid, right):
    left_sum = -10**18
    s = 0
    for i in range(mid, left - 1, -1):
        s += arr[i]
        left_sum = max(left_sum, s)

    right_sum = -10**18
    s = 0
    for i in range(mid + 1, right + 1):
        s += arr[i]
        right_sum = max(right_sum, s)

    return left_sum + right_sum


def max_subarray(arr, left, right):
    if left == right:
        return arr[left]

    mid = (left + right) // 2

    return max(
        max_subarray(arr, left, mid),
        max_subarray(arr, mid + 1, right),
```

```
            max_crossing_sum(arr, left, mid, right)
        )


    T = int(input())

    for _ in range(T):
        N = int(input())
        arr = list(map(int, input().split()))
        print(max_subarray(arr, 0, N - 1))
```

```
1
9
-2  1 -3 4 -1 2 1 -5 4
6
```