Algorithm Design Laboratory with Applications

Prof. Stefano Leucci

Problem: *Mountain trip.*

A road is n kilometers long and passes through several cities. Each city can be either a mountain city or a sea city. There are M mountain cities, the i-th of which is located m_i kilometers after the beginning of the road. Similarly, there are S sea cities and the i-th sea city is located s_i kilometers after the beginning of the road (m_i and s_i are integers between 0 and n, endpoints included, and each kilometer of the road can traverse at most one city).

A travel agency offers T possible trips. The i-th trip starts from kilometer b_i and ends at kilometer e_i of the road, visiting all the cities in-between (endpoints included). Alice wants to buy a trip that visits the largest number of mountain cities and that does not visit any sea city.

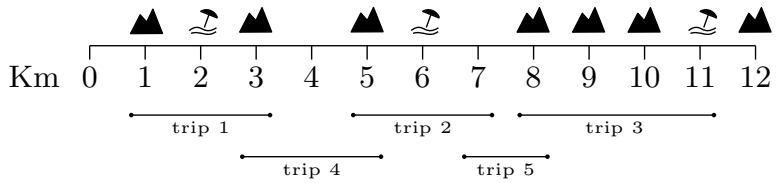
Your task is to design an algorithm that finds the best trip for Alice.

Input. The input consists of a set of instances, or *test-cases*, of the previous problem. The first line of the input contains the number C of test-cases. The first line of each test-case contains the four integers n, M, S, and T. The second line contains M integers, where the i-th integer is the position m_i of the i-th mountain city. The third line contains S integers, where the i-th integer is the position s_i of the i-th sea city. The i-th among the next T lines describes the i-th trip and contains the 2 integers b_i and e_i.

Output. The output consists of C lines, where the i-th line is the answer to the i-th test-case and contains the maximum number η of mountain cities that can be visited by a trip that does not visit any sea city. If all trips visit at least one sea city, then $\eta = -1$.

Assumptions. $1 \leq C \leq 10$; $1 \leq n < 2^{31}$; $0 \leq M, S < 2^{16}$; $0 < T \leq 2^{18}$.

Example.



Input (corresponding to the instance depicted above):

```
1
12 7 3 5
10 8 5 3 9 1 12
6 2 11
1 3
5 7
8 11
3 5
7 8
```

Output:

```
2
```

Requirements. Your algorithm should require $O\left((M + S + T)\log(M + S)\right)$ time (with reasonable hidden constants).

Notes. A reasonable implementation should not require more than 1 second for each input file.