Moonlit Trees

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

A tree by the river can be described as a form with x nodes numbered $1 \sim x$:

- There is exactly one root, numbered 1.
- For each node i ($i \neq 1$), it has exactly one parent numbered f, satisfying $f < i$.

Little B cannot remember such an abstract concept as a "tree", so he invented two ways to turn a tree into a permutation:

Algorithm 1 Transfer Tree to Permutation

Require: sons of each node *son*, interger *type* equals to 1 or 2.

Ensure: queue q.

function DFS(u)
    q.push (u)
    if $type = 1$ **then**
        for each v in $son[u]$ in increasing order **do**
            DFS (v)
        end for
    else
        for each v in $son[u]$ in decreasing order **do**
            DFS (v)
        end for
    end if
end function
DFS (1)

- **Method 1**, $type = 1$: Start from the root, visit the parent first, then the children, and children are visited in increasing order of their numbers.
- **Method 2**, $type = 2$: Start from the root, visit the parent first, then the children, and children are visited in decreasing order of their numbers.

It can be proved that after visiting all nodes, the elements in q form a permutation of $1 \sim x$.

A permutation is called **ambiguous** if it can be obtained from some tree t_1 by method 1 and also from some tree t_2 by method 2. Note that the two trees may be identical, i.e., t_1 may equal t_2.

The bugcat only remembers the positions of $1 \sim n$ in a permutation a of length $n + m$, and asks you to determine, among all possible permutations, how many are ambiguous.

Input

The first line contains an integer T ($1 \leq T \leq 10^5$) indicating the number of test cases.

For each test case, the first line contains two integers n and m ($0 \leq m \leq 80$, $1 \leq (n + m) \leq 10^5$).

The second line contains $n + m$ integers $a_{1\sim n+m}$ ($0 \leq a_i \leq n$). If $a_i = 0$, it means that the bugcat does not remember the value at position i. It is guaranteed that each number from 1 to n appears exactly once in a.

It is also guaranteed that $\sum(n+m) \le 3 \times 10^5$.

Output

For each test case, output a single integer representing the number of ambiguous permutations, modulo 10^9+7.

Example

standard input	standard output
3 3 0 1 2 3 4 1 1 2 3 0 4 3 5 1 2 3 0 0 0 0 0	1 1 109

Note

For the second sample, the only possible permutation is $1, 2, 3, 5, 4$. It can be obtained from the first tree by method 1, and also from the second tree by method 2, thus it is ambiguous.