Weird Ordering (ZCO 2020, Problem 1)

Consider the following algorithm

```
order(arr, i) {
   if length(arr) <= 1 {
     return arr
   }
   l = []
   r = []
   n = length(arr) - 1
   for j in 0, 1, ..., n {
      if ( (arr[j] modulo power(2,i+1))  < power(2,i) ) {
         append arr[j] to l
      }else{
         append arr[j] to r
      }
   }
   l = order(l, i + 1)
   r = order(r, i + 1)
   c = concatenate(l, r)
   return c
}
```

Note that $concatenate(l, r)$ returns an array which is the array l, followed by the array r. Similarly $power(x, y)$ returns x^y.

Let a be the array $a_0, a_1, a_2, a_3, \ldots, a_n$ where $a_j = j$ for each index j and the last index $n = (2^p - 1)$ for a fixed integer parameter p. Given an integer p and an index idx, your task is calculate the element at index idx in the array returned by executing $order(a, 0)$.

For example, suppose \$ p = 3\$ and $idx = 3$.

- The initial array is $a = [0, 1, 2, 3, 4, 5, 6, 7]$.
- Executing $order(a, 0)$ first creates two new arrays $l == [0, 2, 4, 6]$ and $r == [1, 3, 5, 7]$.
- Next, $order(l, 1)$ and $order(r, 1)$ are executed.
- $order(l, 1)$, in turn, executes $order([0, 4], 2)$ and $order([2, 6], 2)$ which return $[0, 4]$ and $[2, 6]$, respectively. These are then concatenated, so $order(l, 1)$ returns $[0, 4, 2, 6]$.
- Similarly, $order(r, 1)$ returns $[1, 5, 3, 7]$.
- These two are concatenated as the final result, so the array returned by $order(a, 0)$ is $[0, 4, 2, 6, 1, 5, 3, 7]$.

So, if the input is $p = 3$, and $idx = 3$, the answer is 6. If $p = 3$ and $idx = 7$, the answer should be 7.

Input Format:

- The first line contains a single integer, T, which is the number of testcases. The description of each testcase follows.
- Each testcase is described by a single line with two integers: p and idx, where p is the parameter that determines the length of the array a and idx is the index at which you have to report the value in the output of $order(a, 0)$.

Output Format:

- You should print the answer in a new line for each testcase, which should be a single integer, the element at index idx after executing $order(a, 0)$ for the array a defined by the parameter p.

Constraints:

- $1 \leq T \leq 10$
- $1 \leq p \leq 50$
- $0 \leq idx \leq 2^p - 1$

SUBTASKS:

- **Subtask 1: 20% points :** $1 \le p \le 20$
- **Subtask 2: 80% points:** Original constraints

Sample Input:

```
2
3 3
3 7
```

Sample Output:

```
6
7
```

Explanation:

Both the testcases have been explained in the problem statement.

Interleavings and Blocks (ZCO 2020, Problem 2)

In an array, a *block* is a maximal sequence of identical elements. Since blocks are maximal, adjacent blocks have distinct elements, so the array breaks up into a series of blocks. For example, given the array $[3, 3, 2, 2, 2, 1, 5, 8, 4, 4]$, there are 6 blocks: $[3, 3], [2, 2, 2], [1], [5], [8], [4, 4]$.

In this task, you are given two arrays, A (of length n), and B (of length m), and a number K. You have to interleave A and B to form an array C such that C has K blocks. Each way of interleaving A and B can be represented as a $0-1$ array X of length $n+m$ in which $X[j]$ is 0 if $C[j]$ came from A and $X[j]$ is 1 if $C[j]$ came from B.

A formal description of the interleaving process is given at the end.

For example, if $A = [1, 3]$ and $B = [3, 4]$, there are 6 ways of interleaving A and B. With each interleaving X of A and B, we also count the number of blocks in the resulting interleaved array C. The descriptions of the interleavings, X, and the outcomes, C, are given below.

- $X = [0, 0, 1, 1]$, which corresponds to $C = [1, 3, 3, 4]$, 3 blocks.
- $X = [0, 1, 0, 1]$, which corresponds to $C = [1, 3, 3, 4]$, 3 blocks.
- $X = [0, 1, 1, 0]$, which corresponds to $C = [1, 3, 4, 3]$, 4 blocks.
- $X = [1, 0, 0, 1]$, which corresponds to $C = [3, 1, 3, 4]$, 4 blocks.
- $X = [1, 0, 1, 0]$, which corresponds to $C = [3, 1, 4, 3]$, 4 blocks.
- $X = [1, 1, 0, 0]$, which corresponds to $C = [3, 4, 1, 3]$, 4 blocks.

Observe that different interleavings X may produce the same array C, such as the first two interleavings in the example above.

Your task is the following. Given arrays A and B and a number K, find the number of different interleavings X of A and B that produce an output array C with exactly K blocks. Note that we are counting the number of interleavings, not the number of different output arrays after interleaving. For instance, if the same output array C is produced via 2 different interleavings, it gets counted twice.

Since the answer might be large, print the answer modulo $10^8 + 7$.

Here is a formal definition of the interleaving process:

Suppose $A = A_1, A_2, ..., A_n$ and $B = B_1, B_2, ..., B_m$. Then, the process of generating an interleaving C can be described using an array X of size $n + m$, with exactly n $0's$ and m $1's$. Suppose we have such an array $X = X_1, X_2, ..., X_{n+m}$. Using this array X, we create the output array $C = C_1, C_2, ..., C_{n+m}$, using the following algorithm:

```
i = 0, j = 0
while( (i+j)<(n+m) )
    if(X[i+j+1] == 0)
        C[i+j+1] = A[i+1]
        i = i+1
    else
        C[i+j+1] = B[j+1]
        j = j+1
```

Thus if the X value is 0, we pick the next available element from A into C, and if it is 1, we pick from B instead. This creates an interleaving of the arrays A and B.

Input Format:

- The first line contains a single integer, T, which is the number of testcases. The description of each testcase follows.
- The first line of each testcase contains three integers: n, m, and K, which denote the size of array A, the size of array B, and the required number of blocks in C, respectively.
- The next line contains n integers, which represent the array A.
- The next line contains m integers, which represent the array B.

Output Format:

- You should print the answer in a new line for each testcase, which should be the number of valid interleaving arrays X which correspond to an output array C with K blocks, modulo $10^8 + 7$.

Constraints:

- $1 \leq T \leq 10$
- $1 \leq n \leq 100$
- $1 \leq m \leq 100$
- $1 \leq K \leq n + m$
- $0 \leq A_i, B_j \leq 10^9$

Subtasks:

- **Subtask 1: 10% points:** $m = 1$
- **Subtask 2: 30% points:** $0 \leq A_i, B_j \leq 1$
- **Subtask 3: 60% points:** Original constraints.

Sample Input:

```
5
2 2 4
1 3
3 4
2 2 3
1 3
3 4
2 2 2
1 3
3 4
2 2 4
4 7
8 5
2 2 2
4 7
8 5
```

Sample Output:

```
4
2
0
6
0
```

Explanation:

- The first three testcases correspond to the example given in the problem statement. Of the 6 interleavings, 4 produce outputs with 4 blocks and 2 produce outputs with 3 blocks. Hence, for $K = 4$, the answer is 4, for $K = 3$, the answer is 2, and for $K = 2$, the answer is 0.
- The fourth and fifth testcases have $A = [4, 7]$ and $B = [8, 5]$. Here are the 6 interleavings of these two arrays.
  - $X = [0, 0, 1, 1]$, which corresponds to $C = [4, 7, 8, 5]$, 4 blocks.
  - $X = [0, 1, 0, 1]$, which corresponds to $C = [4, 8, 7, 5]$, 4 blocks.
  - $X = [0, 1, 1, 0]$, which corresponds to $C = [4, 8, 5, 7]$, 4 blocks.
  - $X = [1, 0, 0, 1]$, which corresponds to $C = [8, 4, 7, 5]$, 4 blocks.
  - $X = [1, 0, 1, 0]$, which corresponds to $C = [8, 4, 5, 7]$, 4 blocks.
  - $X = [1, 1, 0, 0]$, which corresponds to $C = [8, 5, 4, 7]$, 4 blocks.

All 6 interleavings produce outputs with 4 blocks, so for $K = 4$ the answer is 6 and for any other value of K, the answer is 0.