Problem Set 7 - Math

Question F

Upon first reading this question, it was pretty clear that we were dealing with something Inclusion Exclusion principal'y, but how it actually applied to the problem, I did not really see on first glance - only until later once I had worked through the problem with a friend that I saw where the inclusion exclusion principle came in. My initial thoughts about the question were largely correct - I had the idea of first trying to find a whole inclusion set that we could then subtract invalid states from to get our final answer.

If we consider each columns first, the number of ways we can allocate integers into it to make it valid (at least one '1') can be given by the number of ways to allocate k numbers, subtracting the number of ways to allocate just the numbers 2-k to it… i.e.

$$k^n - (k - 1)^n$$

Now, since there are n columns, the ways to allocate n columns that way is:

$$(k^n - (k - 1)^n)^n$$

However, some of those might be invalid, as the rows may end up in a way that ends up invalid, so we need to invalidate (subtract from) some of these arrangements. The way I thought of doing this was trying to find expressions for the number of arrangements where 1 row ends up invalid, 2 rows end up etc.

Consider the case where there are i rows that end up invalid, what is the number of arrangements. The number of combinations of rows that can end up invalid can be given by nCi, and for each of these combinations, we can partition our calculation into the invalid and valid rows. For each column, we need to fill i row positions with invalid arrangements (cannot choose a 1), and we need to fill n-i row positions with valid arrangements (have to choose at least a 1). Using similar logic to the above, the number of ways to choose i invalid rows are:

$$nCi \ * \ (k - 1)^i \ * \ (k^{n-i} - (k - 1)^{n-i})$$

It took me some time to realise though that the valid sections of each of these columns that we are allocating may still form invalid rows, which means that our case for i = 1 doesn't actually mean EXACTLY one row is invalid, it's rather, at LEAST one row is invalid. This at first, implied that we could just subtract the case where i = 1, as it represents cases where at least i rows are invalid, and should encompass all invalid cases, but upon asking a friend about this, I finally realised that this is where the inclusion exclusion stuff comes in.

For the case where at least 1 row is invalid, since the other rows (not forced to be invalid) may still form invalid sequences, we might end up double counting cases. Let's say we're forcing 1 to be invalid, 2 may also end up being invalid. Now when we force 2 to be invalid, 1 may end up being invalid. Thus we are double subtracting the two invalid row cases, and even triple subtracting the three invalid row cases. This would require us to add back the two invalid row cases, and then subtract the overcounted three invalid row cases from that, then add back 4, subtract 5 etc… until the n rows invalid case.

After applying this inclusion exclusion principle, we will have subtracted the sets where exactly 1 - n rows are invalid exactly once each, without double counting - and we thus have our final answer!