Code Jam 2018 - Round 2

Analysis: Costume Change

Test set 1

Trying every possible costume on every dancer will be too slow, even for this test set. Therefore, we need another solution.

To solve this test set, we can first observe that this problem is equivalent to the following: Find the largest subset of dancers such that no two dancers in the subset have the same costume type and share the same row or column.

If we find such a subset, we can change the costume types of the dancers not in the subset so that they follow the rules. We can iterate through the dancers in any order (e.g. row-major). If the current dancer is in the subset, we leave the costume as it is. Otherwise, we pick a new costume type for that dancer. Since a dancer can be in the same row or column as at most $2\mathbf{N} - 2$ other dancers, and since there are $2\mathbf{N}$ costume types, it will always be possible to find a valid type to change to.

So, we can iterate through every subset of of dancers and check whether there are two dancers in the subset having the same costume type and sharing the same row or column. This solution runs in $O(2^{\mathbf{N}^2} \times \mathbf{N}^2)$ time.

Test set 2

To solve this test set, we need a much more efficient way of finding the subset mentioned above. We can solve the problem independently for each costume type. Let f(x) be the largest subset of dancers who are wearing a costume with type x, such that no two dancers in the subset share a row or column. Then the size of our desired subset will be Σ f(i) for $-\mathbf{N} \le i \le \mathbf{N}$, $i \ne 0$.

How can we find f(x)? Let us create a bipartite graph as follows:

- $\{A_i\}$ is a set of vertices, where each vertex corresponds to a row.
- $\{B_i\}$ is another set of vertices, where each vertex corresponds to a column.
- If and only if the dancer in the i-th row and the j-th column wears a costume of type x, we add an edge connecting A_i and B_j.

To find the largest valid subset of dancers, we need to find the maximum independent set of this graph, since we can only pick one dancer for each row and column. We can use a [maximum cardinality bipartite matching algorithm](https://en.wikipedia.org/wiki/Matching_(graph_theory)#In_unweighted_bipartite_graphs) to solve this problem.

A simple maximum cardinality bipartite matching algorithm runs in O(VE), where V is the number of vertices and E is the number of edges. Since V = $O(\mathbf{N})$ and the sum of E among all costume types is $\mathbf{N}^2$, this solution runs in $O(\mathbf{N}^3)$, which is fast enough to solve test set 2.