Code Jam 2015 - World Finals

Analysis: Merlin QA

Every spell in the spell-testing procedure can be interpreted as an **M**-dimensional vector, and the current state of Edytheâ€™s inventory is also an **M**-dimensional vector (initially an all-zero vector). Casting a spell is equivalent to adding the vector describing the spell to the vector describing Edytheâ€™s inventory, and then changing all the negative values in Edytheâ€™s inventory to zero (this corresponds to retrieving the exact necessary amount from Merlinâ€™s storehouse). Edythe is aiming to maximize the sum of all the coordinates at the end.

Note that even if we allowed Edythe to zero a coordinate of the vector describing her inventory at any time, it would never be advantageous to zero it when it is positive, and it would never be disadvantageous to zero it when it is negative - so the final answer will not change if we give Edythe full freedom over when and which coordinates of her inventory get zeroed. We will solve this less constrained problem from now on.

Notice that it never makes sense to zero a given coordinate more than once, since the last zeroing will make the previous ones irrelevant anyway. So from now on, we will solve an equivalent problem — given a list of spells, and the opportunity to zero each coordinate of the accumulated sum *exactly once, at any point of the summation*, maximize the final sum of all the coordinates. Thus, a solution candidate can be equivalently described as a permutation of the **N** spells, plus the moments in time at which we zero each coordinate.

Letâ€™s fix the order in which coordinates get zeroed (we will later iterate over all **M**! possible orderings), and look at one particular spell. We will try to figure out the best time to cast it. If we cast it before any coordinate is zeroed, then the result will be the same as if we had not cast it at all — every coordinate will eventually be reset to zero after casting the spell anyway. Now assume we cast it when *k* coordinates have already been zeroed, and the others have not. Compared to the situation when we donâ€™t cast it at all, the end result is that the values of the spell on the *k* already-zeroed coordinates will be added to the final values of these coordinates, and so the sum of these values will be added to Edytheâ€™s final result. So, for every spell, we check each of the **M**+1 possibilities for k, and choose the one which will give the largest contribution to the final result.

So a solution can work as follows: take every possible ordering **P** of the **M** dimensions. For each ordering P, iterate over all the spells, and for each spell find which of the **M**+1 values for k will cause the spell to contribute the most to the final result. The sum of all these contributions for all the spells is the best possible score for this ordering **P**; the maximum score over all orderings is the answer. The runtime is O((**M**+1)! **N**), which will easily run in time for **N** = 100 and **M** = 8.