Problem Tutorial: “K-onstruction”

Consider a set S of integers, in which there are exactly K subsets with sum 0, in which there are no zeros, and in which sum of all elements is not zero. Let P and N be the sums of all positive and of all negative elements of the array correspondently, wlog $P > -N$. Let's add some nonzero elements divisible by P to this set, denote the set of these added elements by T for now.

Let's look at $S \cup T$. How many subsets with zero sum are there in it? The part we take from T is divisible by P, so from S we also have to take part divisible by P. As $P > -N$, there are only 2 ways to do so: to take sum P by choosing all positive elements (in exactly one way), or to take sum 0 in K ways.

So, the number of subsets with sum 0 in $S \cup T$ is equal to $K \times$(number of subsets with zero sum in T) + (number of subsets with sum $-P$ in T). Note that the set $S \cup T$ also satisfies the conditions for S: all elements are nonzero, and sum of all elements is not zero (as it's not divisible by P).

Now, let's generate some small sets S and see what pairs (number of subsets with sum 0, number of subsets with sum 1) they produce. If for set of size n there are cnt_0 subsets with sum 0 and cnt_1 subsets with sum 1, we have a transition from (len, K) to $(len + n, cnt_0 K + cnt_1)$.

Based on these generated transitions, calculate dp array, where $dp[n]$ denotes the smallest length needed to get exactly n subsets with sum 0, and save the info by which transitions we should go to.

It turns out that generating all sets with integers from $\{-3, -2, -1, 1, 2, 3\}$ with size at most 10 is enough to make all values of dp up to 10^6 less or equal to 30, and this fits without any optimizations. We can also fit 29 easily, and will have to optimize quite a lot to fit into 28, so we decided to make the bound on size of the array 30.