Code Jam 2015 - Round 3

Analysis: Log Set

Let x be the largest element of S'. Then x is the sum of all of the positive elements of S, and none of its negative elements.

Let y be the second-largest element of S'. Either this is the sum of all but the smallest of the non-negative elements, or it is the sum of all of the non-negative elements and the smallest negative element. So if d = x-y, then the smallest magnitude of any element in S is d — either d $\in$ S or -d $\in$ S.

These two cases are basically identical. Consider the process of building every subset of S, and finding the sum of each of those subsets. To build a subset, we consider each element of S, and either include it in the subset or not. If we include it, the sum increases by the value of the element. If we don't include it, the sum increases by zero. So in the case where d is an element of S, we will be choosing between adding 0 and adding d to the sum. If -d is an element of S, we will be choosing between adding -d and adding 0. These situations are equivalent, except that every element of S' is offset by a constant depending on whether d or -d is in S.

We can now scan through S' and match elements into pairs that are d apart, and keep only the smallest of each pair. Then we can recursively apply this procedure to get the magnitudes of all the other elements of S.

When we are done, we have the magnitudes of the elements of S. Next we need to figure out which elements are positive and which are negative. First, make them all negative. The largest element of S' will be zero in this case. Then we need to choose which elements to change from negative to positive in order to make the largest element of S' equal to x. This is a Subset Sum problem, with required sum x, and elements equal to the magnitudes. This is easily solvable with dynamic programming — the number of unique sums, and hence the number of states in the dynamic program, is guaranteed to be small because the number of unique values in S' is small.