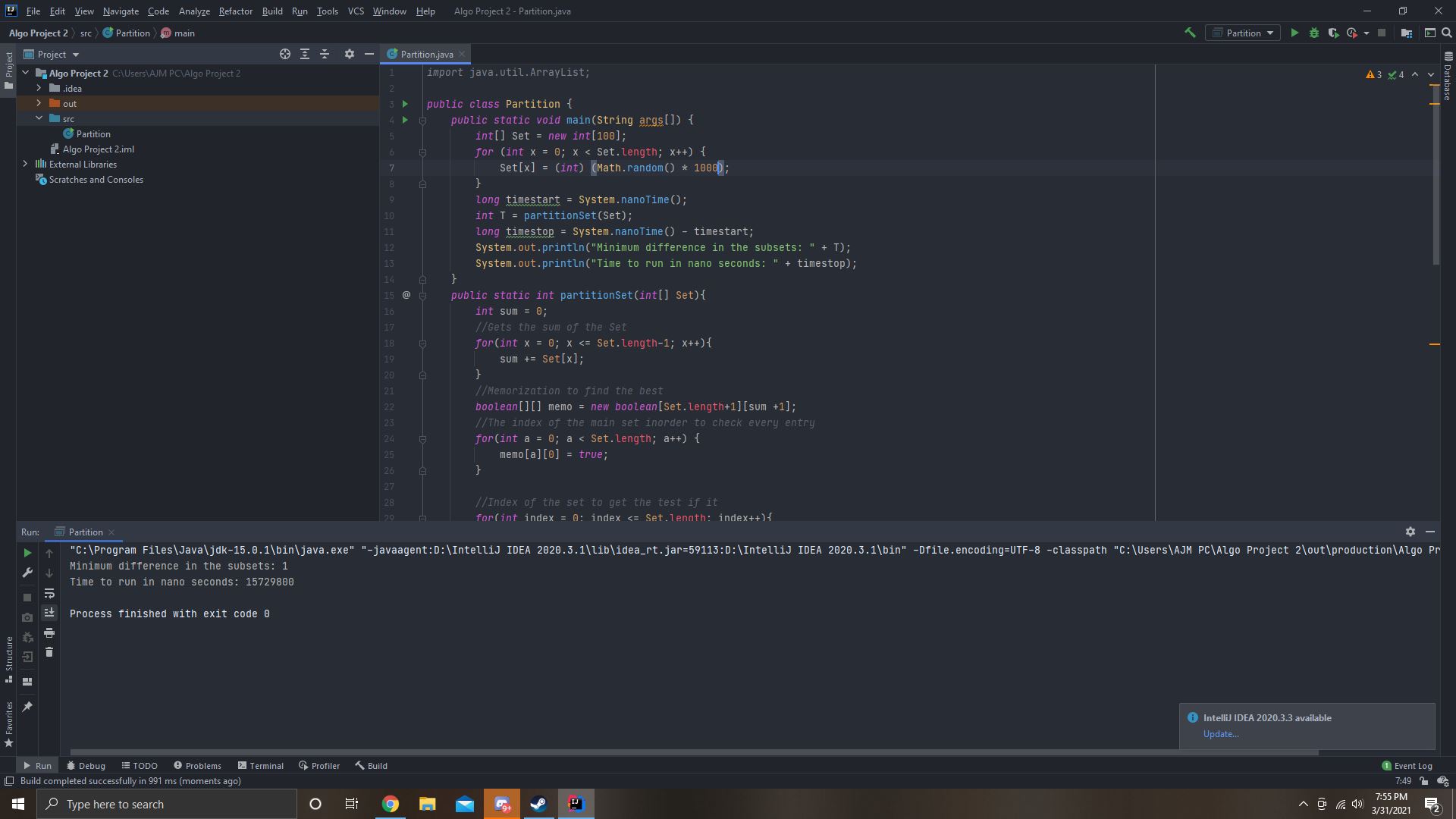
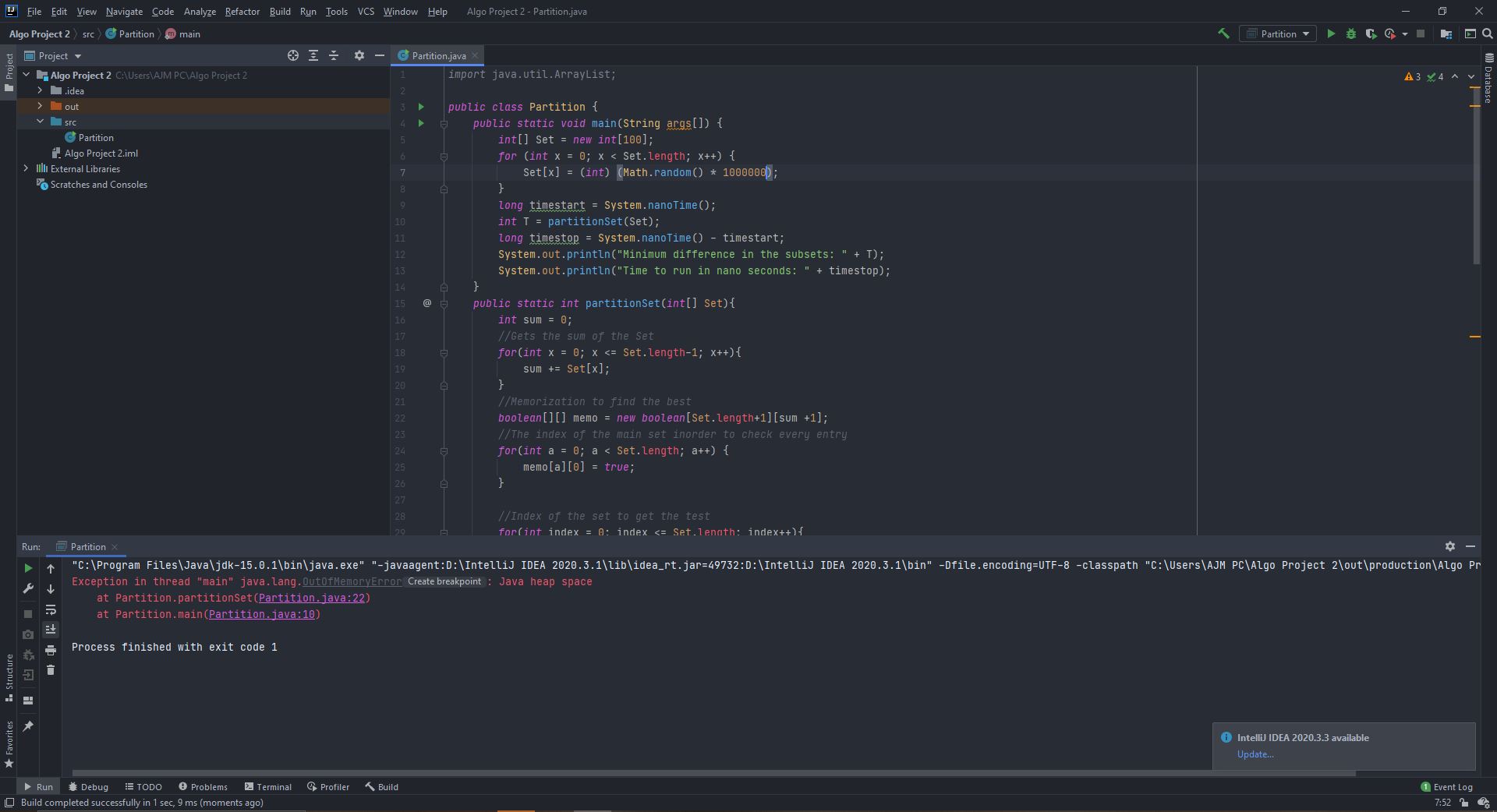
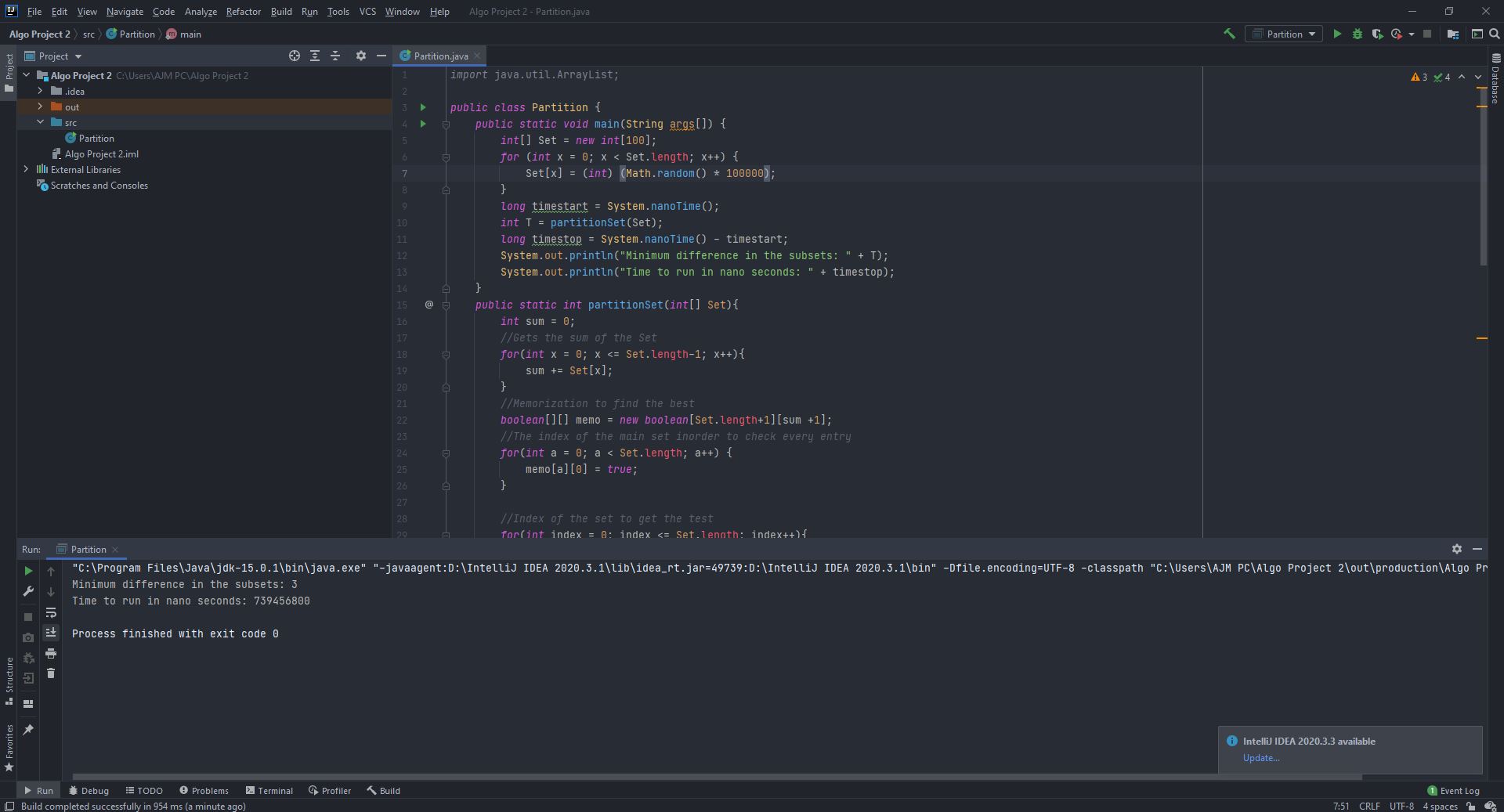
Test will 1,000:



Test with 1,000,000: I kept got a memory error(Java heap error)



Test with 100,000: Which didn’t give errors



Formula:

S1[index][subsetsum] = {True (S1[index-1][subsetsum],S1[index-1][subsetsum-S[index-1] ) if subsetsum <= sum/2}

= {S1[index-1][subsetsum] else}

Reasoning:

This formula checks attempts to fill the first subset with the entries that will get the sum of the subset to be as close to or equal to half the sum of the entire set as possible. The second subset will get the remaining entries, if the first subset is equal to half then the second will also be equal to half. If not the first will attempt to get the best subset to be as close as possible to half, This means that the second subset will be using the remaining which should be the closest as it can be.

The way that it gets the best Subset is by checking if the current index can is less than the subset sum. Then it checks to see if the last subset version without the new entry is valid or if the last subset that used the current entry is a valid subset. If they are included and consider valid or neither are valid then it would be excluded. If the entry is greater than the current subset sum than the is excluded from the subset and moves on.

The difference in sums is calculated by finding the first subset that has gone through the main set, is less than half and true(A subset less than half the sum at the current index). Then multiplying it by two because the amount less than the half subset 1 is also the amount more than the half that subset 2 should be. These numbers are low but some partitions can have much higher.lf