Assigment 4: Time Efficiency Analysis

The algorithm first checks if d >= dailyExpediture.length (which is n), if it is, it will throw IllegalArgumentException.

Then a for loop goes from 0 to dailyExpediture.length-d.

Within the for loop is another for loop that assigns the elements of prior days expenditure to an array, and then quicksort the array.

Then we find the median of the array elements and compare it to the day’s expenditure we want to compare. If it is >= 2\*median, we increase the numberOfFraud count.

Return the numberOfFraud count at the end of the outer for loop.

The outer for loop of getNumberOfFrauds method runs n - d times.

The nested for loop to get priorDaysExpeditures array runs at most d\*n times.

The quickSort method will have at most d\*logd\*n comparisons.

All the other statements are just assignment statements runs at O(1);

d\*logd\*n dominates the run time, so the algorithm is O(d\*logd\*n).