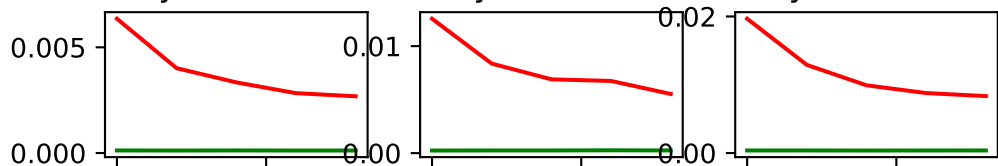
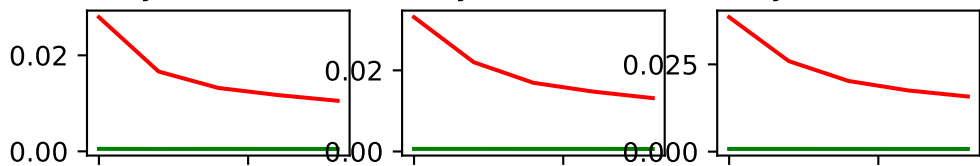


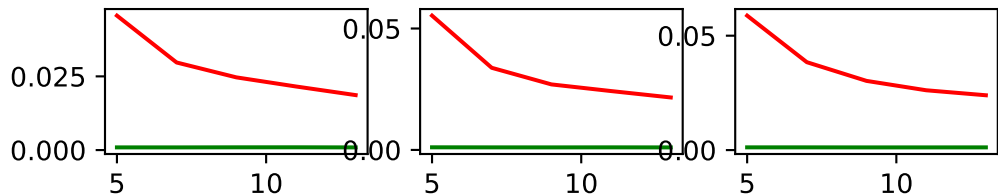
Array Size = 1000 Array Size = 2000 Array Size = 3000



Array Size = 4000 Array Size = 5000 Array Size = 6000



Array Size = 7000 Array Size = 8000 Array Size = 9000



Red line = the deterministic select algorithm

Green line = the sort and pick algorithm.

As we can see from the graph, the sort and pick algorithm outperforms deterministic select algorithm. Also, as m gets larger, the running time for finding the k th smallest number is decreasing. This tells us that the optimal value of m is not in the range of $\{5, 7, 9, 11, 13\}$. We expect the running time to decrease as m gets larger. It will get larger until it hits the length of the array. This means that the optimal value for m is basically the length of this array. Computing the deterministic select algorithm for such m is, in fact, sort and pick algorithm. We obtain such a result because for the larger size of an array, we will need to sort a large number of smaller arrays of size m . This accounts for the increasing time for deterministic select algorithm.