Technical Behavior of the Algorithms :

When both the x axis (no of objects) and y axis (time taken) is converted into log scale of base 2 we end up with the above shown graph. We observe the slope for the divide and conquer is more than that of the normal dynamic programming with trace back. We can see that the time taken for divide and conquer is more than that of normal dp.

In Divide and Conquer method we will have recursive calls and calculations making the running time for the same number of objects more but the trade of here is the number of objects it can process. In my system, normal DP with traceback was able to calculate up to 4096 but the number of objects for dive and conquer is > 16384… this is because we are only using a 2 column 2D array saving space in the heap.

One thing that is common for all the methods is that time taken increases with the increase of number of objects. This not much of a shock since more number of objects = more calculations = more time.

Key points we can get from the graph :

1. Divide and conquer is able to calculate for more than 16384 number of items but normal DP stops at 4096. Cannot go more than that because of the heap overflow.

2. The time taken for the divide and conquer is more because of the recursive and over head calls , nomal dp does not have the recursive calls so its fast but space dependent.

This is a simple trade of between space and time.