We can see from the graph that the runtime of both the Recursive and Memoizing algorithms remains approximately the same throughout the increment of n from 10 to 200, while m = 50, and l1,l2 = 100.

The memozing algorithm performs about the same as the recursive algorithm.

In both approaches, at n = 40, the execution time is much higher.

----------------------------------------------------------------------------------------------------------------------------------------

**Questions asked in the assignment: (not sure if these are required, so I added them anyway)**

n = object count

m = bound of the objects

l1 = first knapsack size

l2 = second knapsack size

Q) At what object count does it become infeasible to run the algorithm?

For both Recursive and Memoizing functions - While keeping m = 50 and l1, l2 = 100, the algorithm performs just as fast at even high values of object count (for example n = 10000). At n = 1000000, the algorithm starts to slow down, taking about 7 times more time. At n = 10000000, the algorithm becomes infeasible.

Q) Repeat the same experiment with memoizing solution knapMemo(). How much faster is the memoizing method?

The memoizing method takes nearly the same amount of time as the recursive method, for random n values ranging from 10-200 , with m = 50 and l1 = 100 & l2 = 100.

Q) Experiment with much larger problem sizes (L1, L2, n and also m). What variable values cause the memoizing algorithm to slow down and why?

m - Runtime is not affected by the value of m in either recursive or memoizing approach. Since m is simply a bound on the range of objects, increasing m does not affect nuumber of recursive calls, so it does not affect the run time of the algorithm.

Since our recursive algorithm terminates by diminishing l1,l2 and n, higher values of either of these creates more recursive calls, slowing down the algorithm.

n - values of 1000000 and higher cause the algorithm to slow down (for both recursive and memoizing approach).

l1 & l2:

Recursive - Values of 10000 or higher cause the algorithm to slow down.

Memoizing - Values of 1000 or higher cause the algorithm to slow down.