

# Lab 1

## Dynamic programming

Recursive (finding maximum value with capacity and  $n$  items)

maximizeRobWorthRecur (capacity,  $n$ , size[], worth[])

//  $n$  is # of items, capacity is total capacity allowed

// base case

if ( $n$  is  $= 0$ , or capacity is  $0$ )  
then,

return 0 // there are no more items or capacity is reached

else if (size of element  $n$  is  $>$  capacity)

return maximizeRobWorthRecur (changing  $n$  to  $n-1$ )

// Since element we are looking at exceeds the capacity we turn to the next element on the left

else

// we check both cases if we include an item or excluded the item.

Then we take the value of the biggest of the two and return it.