Question: https://leetcode.com/problems/integer-break/

Since we have n>=2 we can consider n-1(1 - (n-1)) possibilities for the first value of integer break, because if we include n then the sum will be n+0 and the product will also be zero.

So we will use a helper function for other integer break values other than the first number of integer break, in that case we will have 1 - n possibilities in helper function.

For helper function we will see that there are repeatation so we can use memorization to optimize the algorithm.

Code:  
class Solution {

HashMap<Integer, Integer> mem = new HashMap<Integer, Integer>();

public int integerBreak(int n) {

int max=Integer.MIN\_VALUE;

for(int i=1; i<n; i++){

max=Math.max(max,i\*helper(n-i));

}

return max;

}

public int helper(int n){

int max=Integer.MIN\_VALUE;

if(n==0){

return 1;

}

if(mem.containsKey(n)){

return mem.get(n);

}

for(int i=1; i<=n; i++){

max=Math.max(max,i\*helper(n-i));

}

mem.put(n,max);

return max;

}

}

Github Link :<https://lnkd.in/ecwtJeaz>