Why Not Total Call Stack Depth?

The total no. of calls is exponential, yes — but: • They don't all exist simultaneously

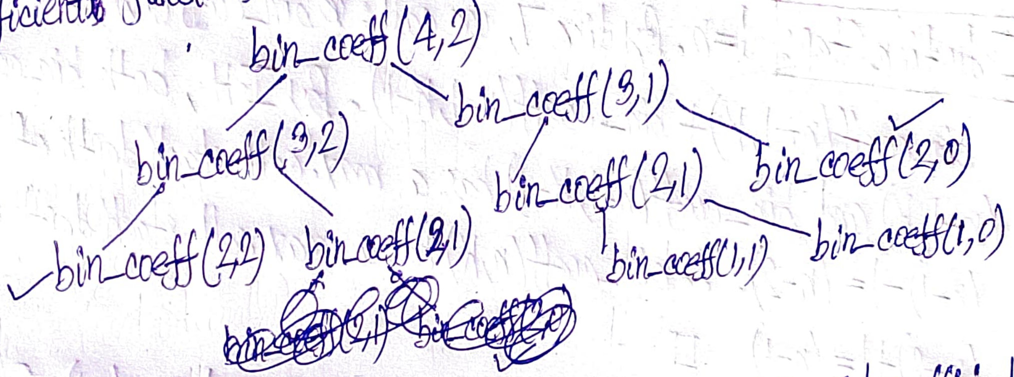
- Once a call returns, its stack frame is popped off.
- Only a single branch is active at any given time (depth-wise)

$\boxed{\text{Space complexity: } O(n)}$

$S_{recursive\_fibonacci}(n) = 16n$ (Not sure)

3) Determine the space complexity of the iterative and recursive binomial coefficients functions created in Exercise 9, Section 1.3

Soln:



Claim: The maximum call stack depth of the recursive_binomial_coefficient function is $K+(n-K-1) = n-1$ for $n > K$

Proof: The proof is by mathematical induction on n.

Base Case: $n = K+1$. We have to prove that the maximum call stack depth for recursive_binomial_coefficient$(K+1, K)$ is K. Again, this will be proven by mathematical induction on K.

Base Case: $K=0$. When $K=0$, we correctly return 1 in line 4. There is 0 call stack depth.

I.H: When $K=l$, the maximum call stack depth for recursive_binomial_coefficient$(l+1, l)$ is $K=l$.

Induction Step: Let, $K = l+1$. We know, bin_coeff$((l+1)+1, l+1) =$ bin_coeff$(l+2, l+1) =$ bin_coeff$(l+1, l+1) +$ bin_coeff$(l+1, l)$ by line 7.