

## Question 2. [5 marks]

Here is a recursive function and a call of that function. What is the value of the call `rec(2,2)`? Please carefully show your work.

```
def rec(a: int, b: int) -> int:
    if a + b <= 0:
        return 1
    return rec(a - 1, b) + rec(a - 1, b - 1)
print(rec(2, 2))
```

## Question 2. [5 marks]

- ▶ We can solve this using a bottom-up or top-down approach.
- ▶ Let's use a bottom-up approach. We can calculate what is returned by calls that are one step away from the base case. Then use those to calculate calls that are two levels from the base case, etc.

## Question 2. [5 marks]

### Level 1:

- ▶  $\text{rec}(1, 0) = \text{rec}(0, 0) + \text{rec}(0, -1) = 1 + 1 = 2$
- ▶  $\text{rec}(0, 1) = \text{rec}(-1, 1) + \text{rec}(-1, 0) = 1 + 1 = 2$
- ▶  $\text{rec}(-1, 2) = \text{rec}(-2, 2) + \text{rec}(-2, -1) = 1 + 1 = 2$

### Level 2:

- ▶  $\text{rec}(1, 1) = \text{rec}(0, 1) + \text{rec}(0, 0) = 2 + 1 = 3$
- ▶  $\text{rec}(2, 0) = \text{rec}(1, 0) + \text{rec}(1, -1) = 2 + 1 = 3$
- ▶  $\text{rec}(0, 2) = \text{rec}(-1, 2) + \text{rec}(-1, 1) = 2 + 1 = 3$

### Level 3:

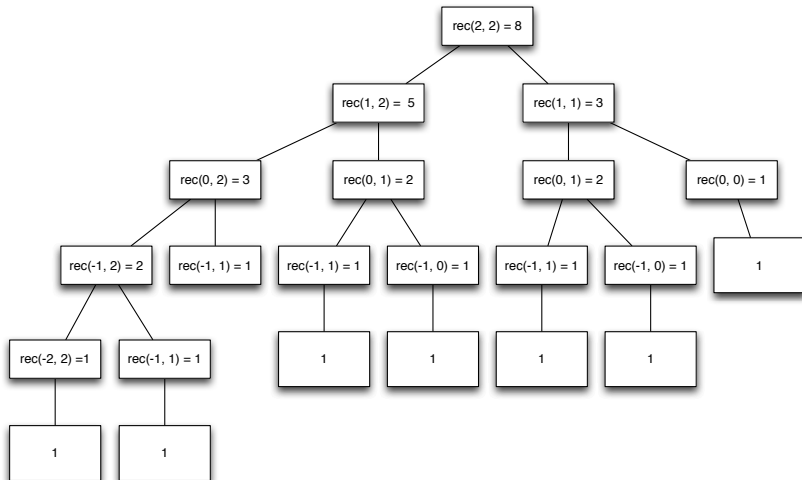
- ▶  $\text{rec}(2, 1) = \text{rec}(1, 1) + \text{rec}(1, 0) = 3 + 2 = 5$
- ▶  $\text{rec}(1, 2) = \text{rec}(0, 2) + \text{rec}(0, 1) = 3 + 2 = 5$

### Level 4:

- ▶  $\text{rec}(2, 2) = \text{rec}(1, 2) + \text{rec}(1, 1) = 5 + 3 = 8$

## Question 2. [5 marks]

We can also use a tree representation to solve this problem.



## Question 2. [15 marks]

Write the body of the function below so that it satisfies its docstring. Your code *must* be recursive.

```
def fib(n):  
    """int -> int  
    Return the nth fibonacci number.  
    Where the fibonacci numbers are defined as:  
    1, 1, 2, 3, 5, 8, 13  
    (each number is the sum of the two previous numbers)  
    >>> fib(4)  
    3  
    """
```

## Question 2. [15 marks]

- ▶ What is the base case?
- ▶ What is the recursive structure?
- ▶ How can we model a Fibonacci sequence mathematically?

## Question 2. [15 marks]

```
def fib(n):  
    """int -> int  
    Return the nth fibonacci number.  
    Where the fibonacci numbers are defined as:  
    1, 1, 2, 3, 5, 8, 13  
    (each number is the sum of the two previous numbers)  
  
    >>> fib(4)  
    3  
    """  
  
    if n < 0:  
        return  
    elif n == 0:  
        return 0  
    elif n == 1:  
        return 1  
    else:  
        return fib(n-1)+fib(n-2)
```