Note that in every function call, we end up making 2 calls.

So, the function calls ends up looking like a tree :

```
                F(0,0)
              /        \
          F(0, 1)       F(1, 0)
           /    \       /      \
       F(0, 2)  F(1,1)  F(1, 1)  F(2, 0)
          ....
```

The function calls end up making a complete binary tree.

```
  Number of calls on Level 0 = 1
  Number of calls on Level 1 = 2
  Number of calls on Level 2 = 4
  ...
  Number of calls on level i = 2^i.
```

```
Total number of calls = 1 + 2 + 4 + ... 2^i + ... 2^(M + N - 2)
                      = O(2^(M + N))
```

Suggest Edits | Close