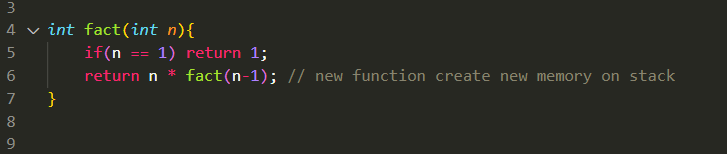
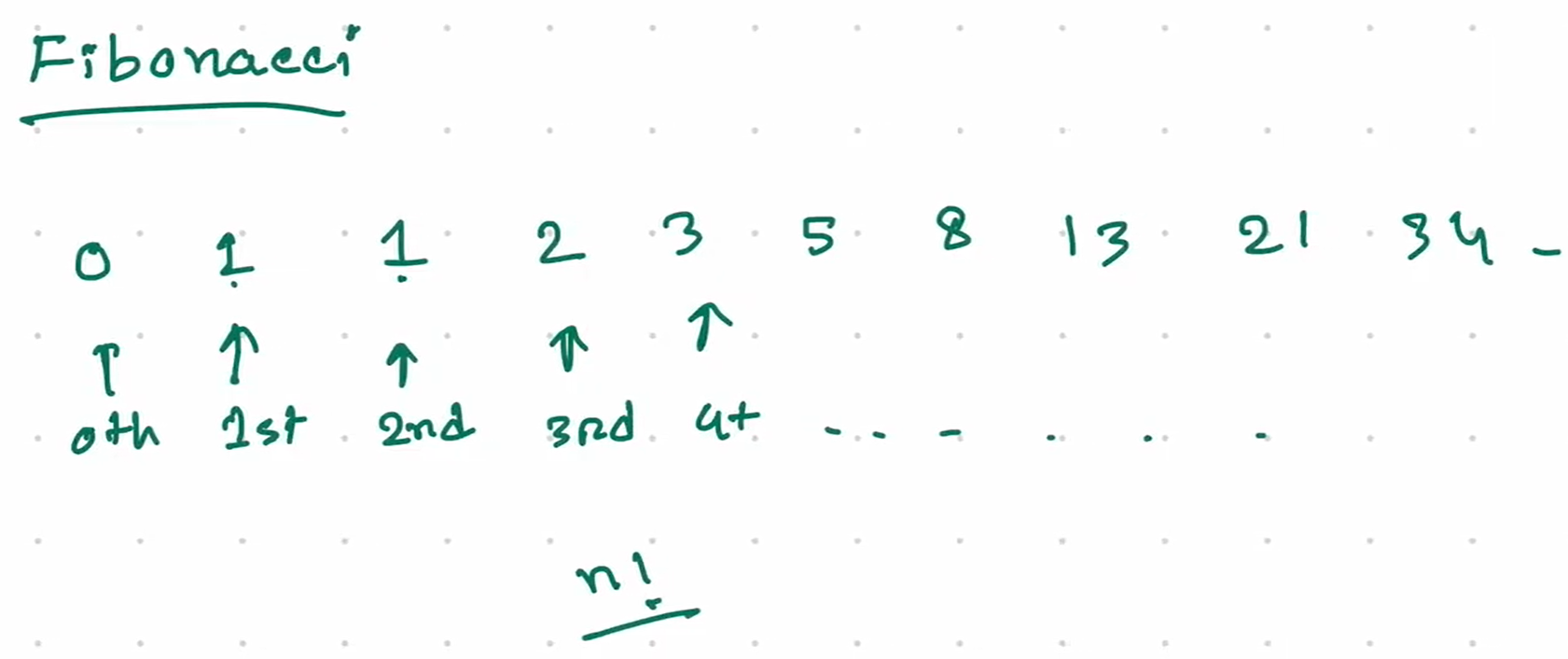
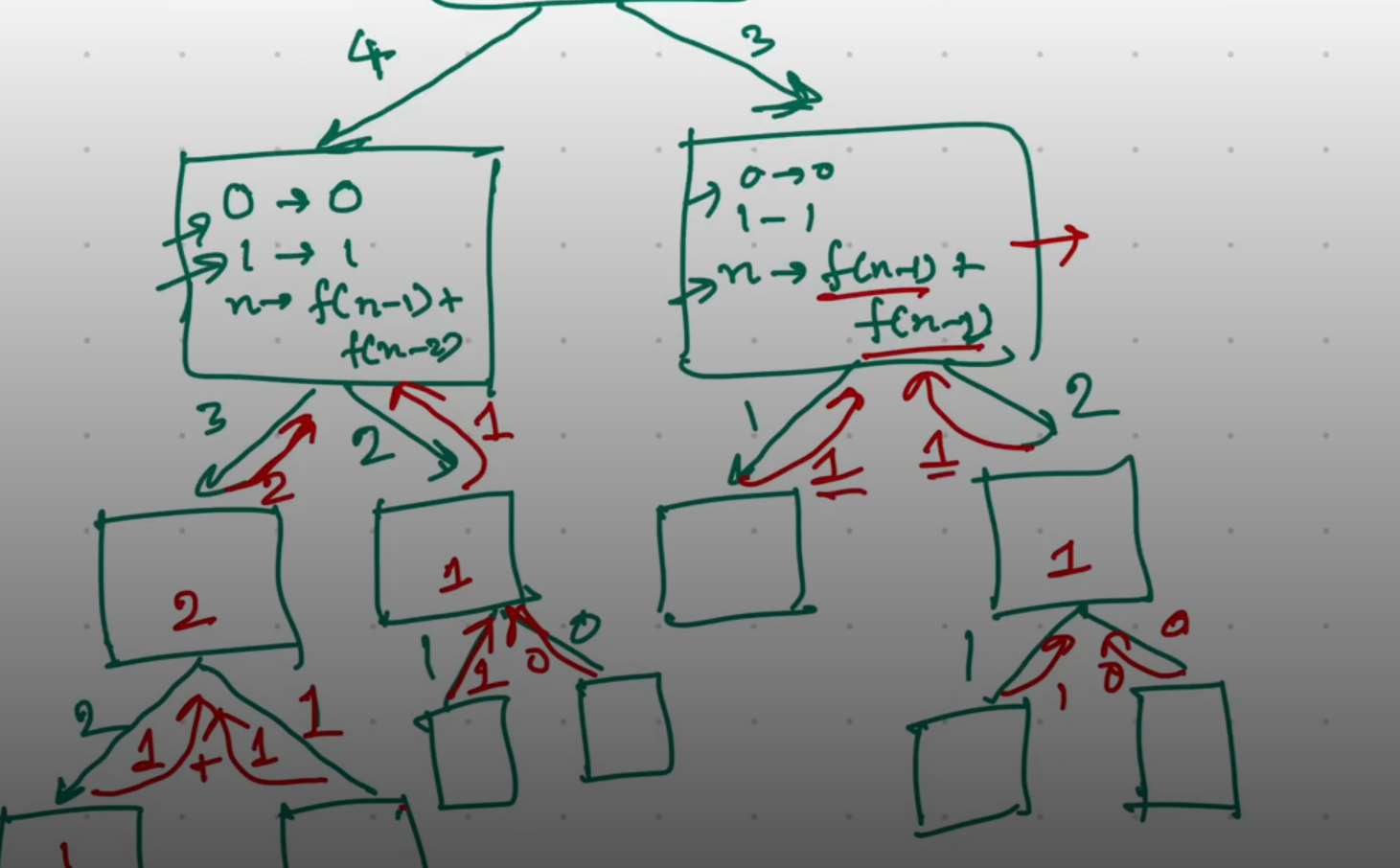


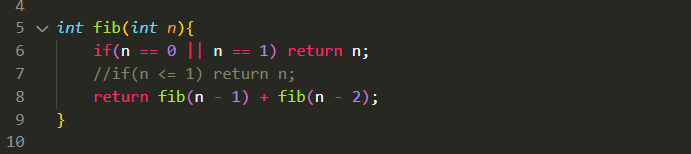
**Implementation of this problem:**

****

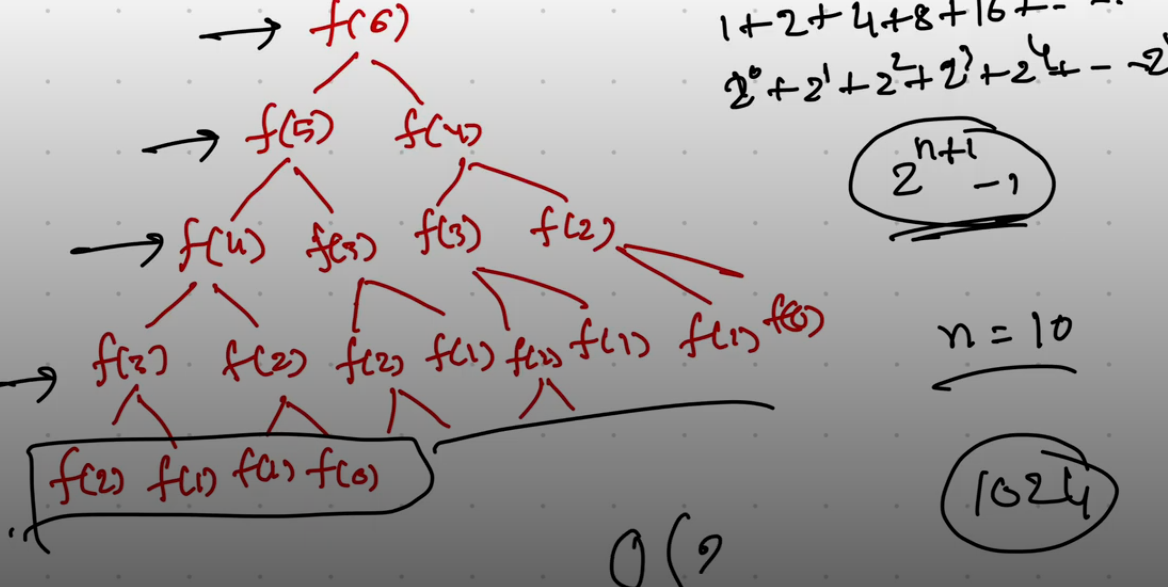




**Implementation:**

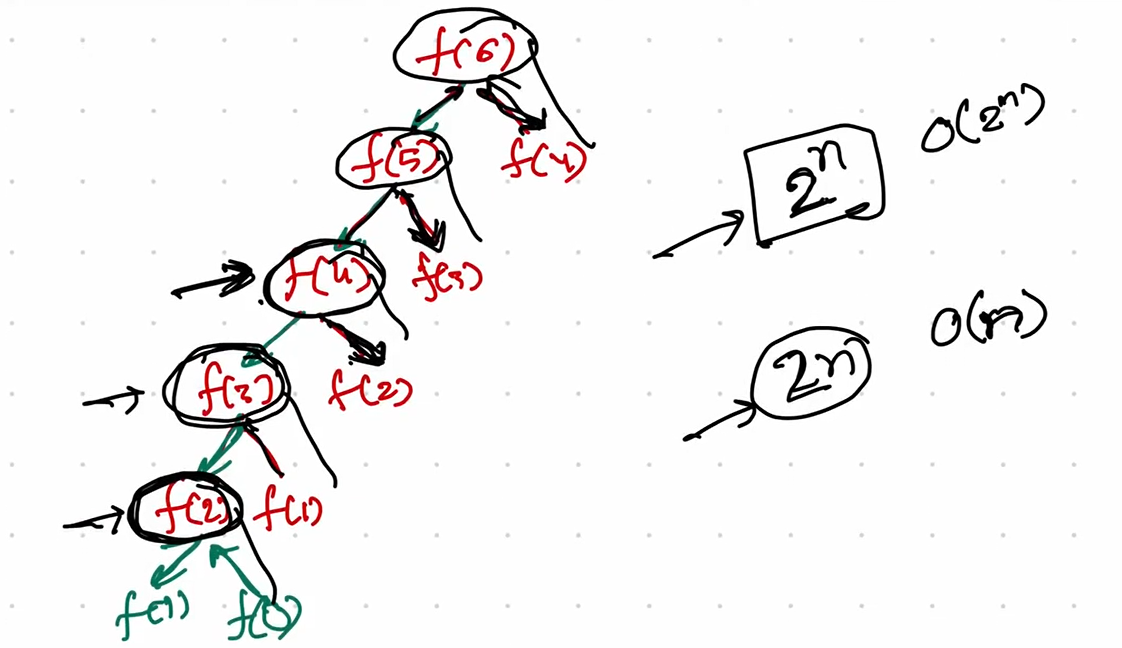


time complexity: O(2^n) // exponential



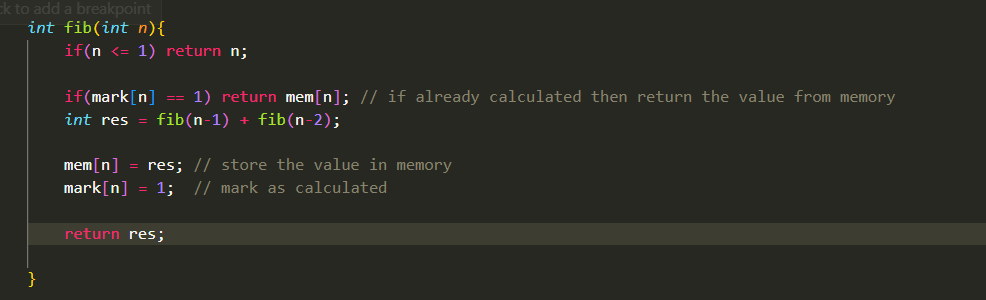
Ekhne kicu function bar bar call hocce, like : f(4), f(3), f(2) and more……jeta complexity baraia diche.Ekbar ber kore store kore rakhle, then again proyojon hole oy store kora value ta use korle complexity reduce kora possible, let’s see..

**Recursion Tree :**



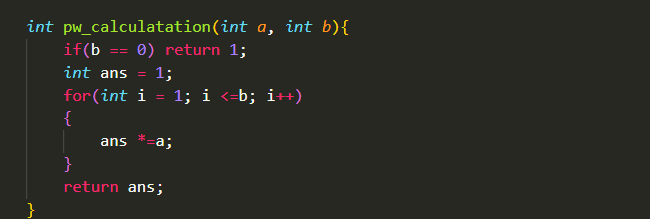
Every function is executing independently, no related to each other, every function allocated new memory.

Optimal Solution will be like this:



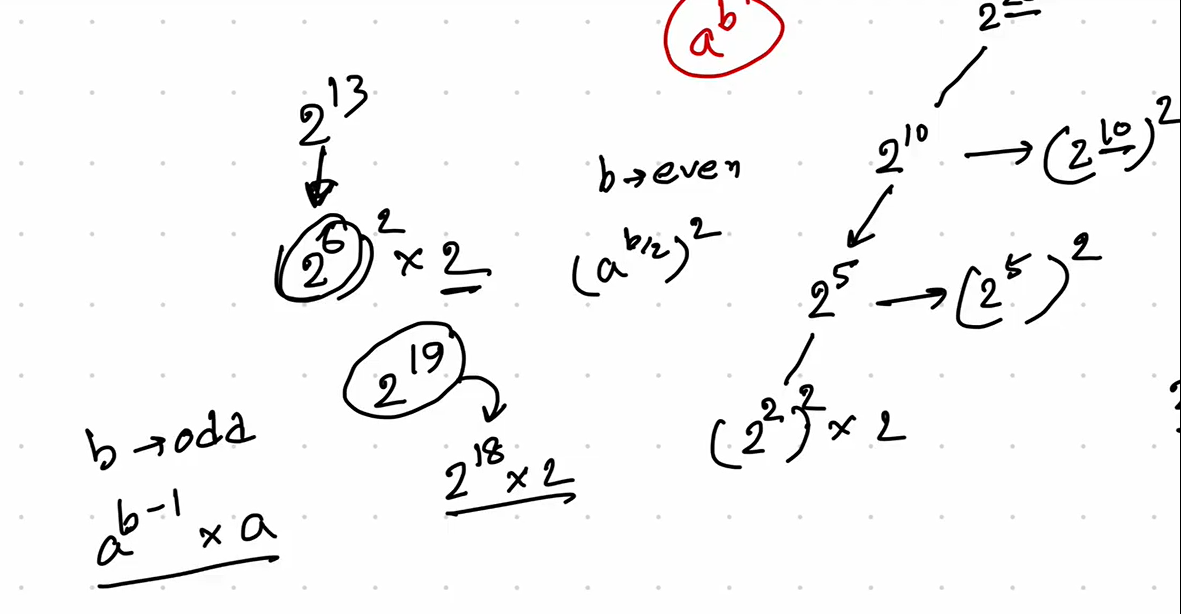
Complexity: O(n)

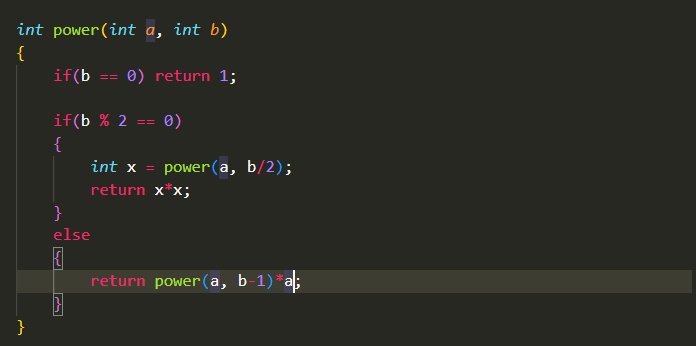
**Question**: given a, b, find out the a^b

Bruteforce solution, without pow function.

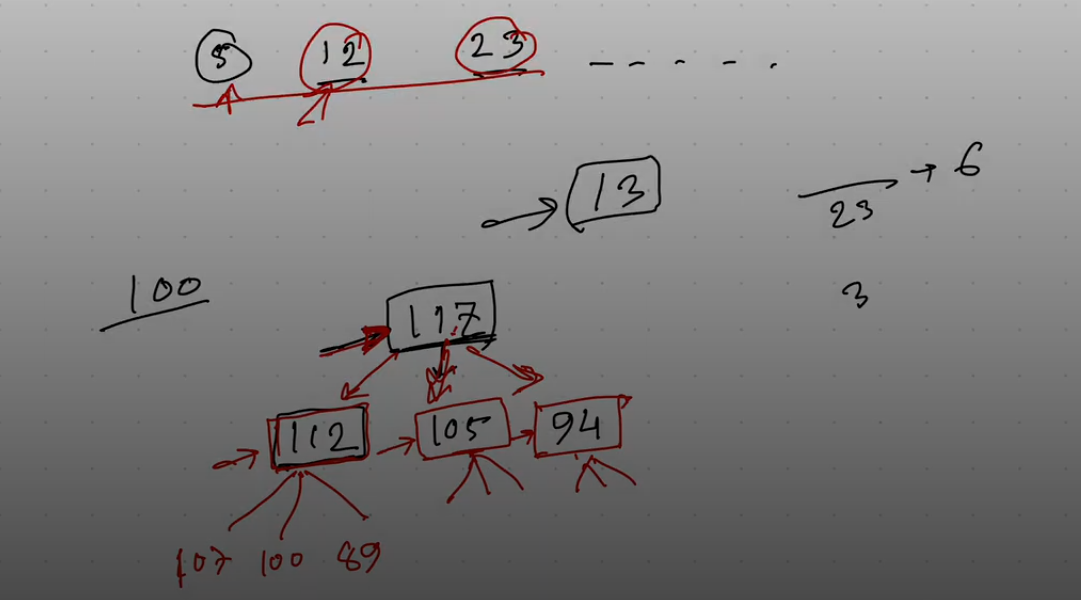
//Complexity: O(b)

Recursive solution visualization:

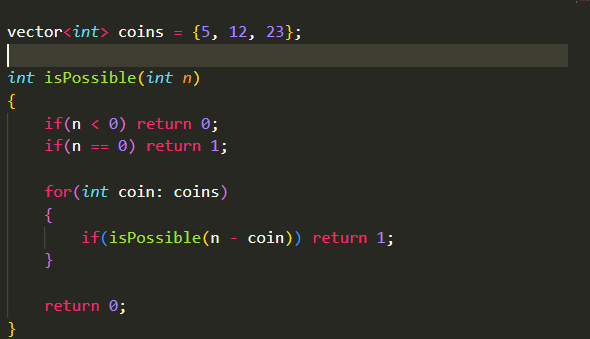




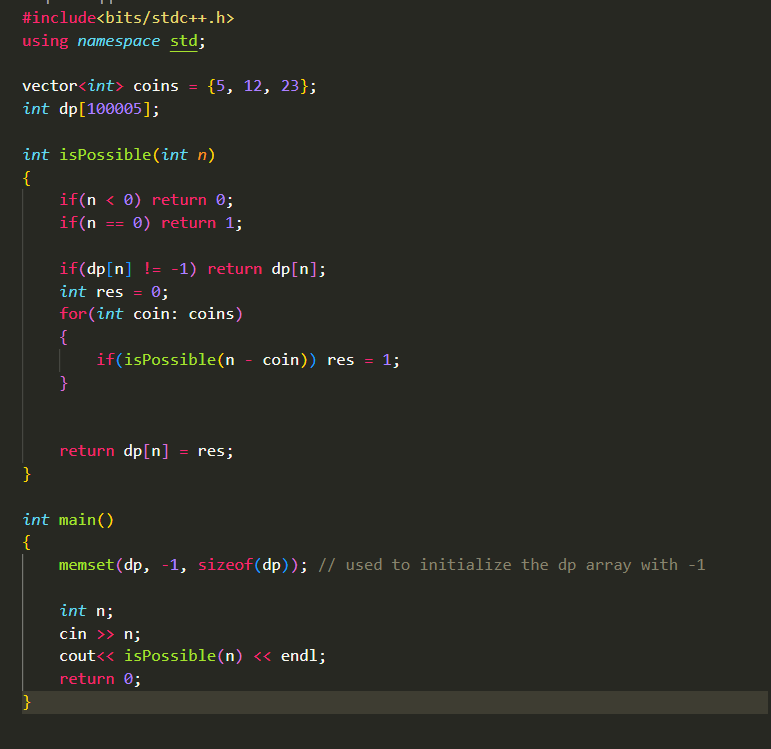
Another problem:



Sol:



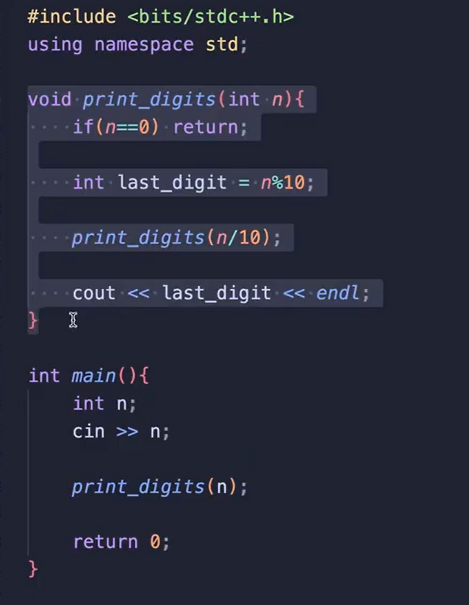
Optimal:



**Question: given an integer, return its all digits like 123 , return 1, 2, 3 seperately**

****

**Optimal:**

****