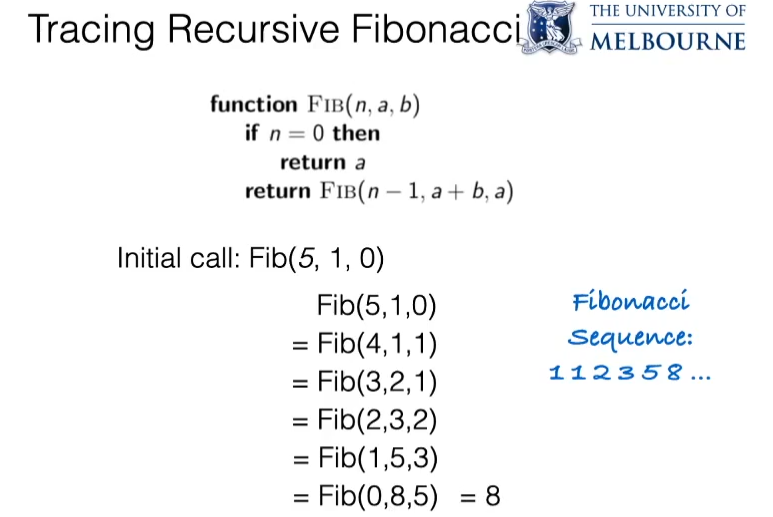
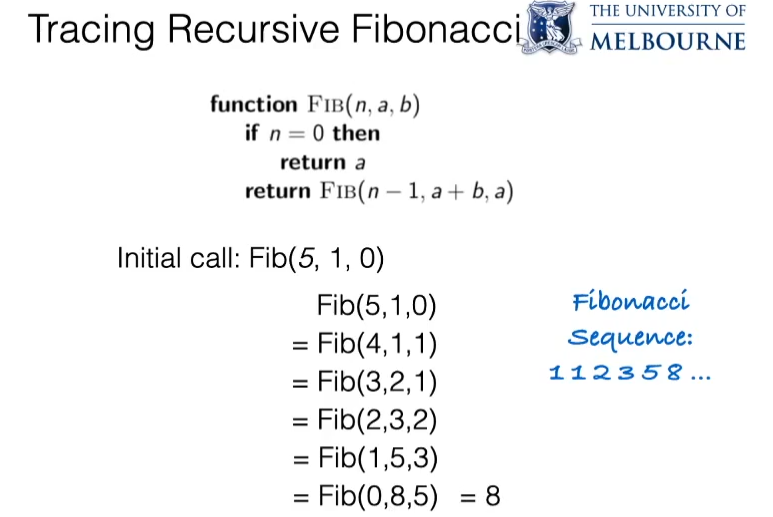
HOW WOULD THE RECURSIVE FIBONACCI WOULD WORK



In recursion the same function gets called over and over until we would reach end of the algorithm.

In above algorithm the main concept would be the in the next recursive call, the previous value of “**a”** the new value of “**b**”.



As we call In the above recursive call on Fib(5,1,0), Initially a would be 1 and b would be zero.

|  |  |
| --- | --- |
| Fib(5,1,0) => Fib(n,a,b) | |
| n | 5 |
| a | 1 |
| b | 0 |
| As the algorithm operates, Initially we would check if n=0 or not which is not true as n=5. So we would not be returning a, instead we would returning FIB(n-1,a+b,a) that would be equal to FIB(5-1,1+0,1) = FIB(4,1,1) = FIB(n,a,b). So from this we can see that, the new value of b in the second call is equal to the earlier value of a which is 1. And we would also have a+b in each next call of recursion. | |
| Next rec.call - Fib(4,1,1) => Fib(n,a,b) | |
| n | 4 |
| a | 1 |
| b | 1 |
| We would check if n=0 or not, which is not TRUE. So we return FIB(n-1,a+b,a) which is the new FIB(n,a,b) in the next call or a+b of previous recursive call is the a of the next call, and also a of the previous call is the b of the next call.  This would return FIB(n-1,a+b,a) or FIB(3,2,1). | |
| Next rec. call – Fib (3,2,1) = Fib(n,a,b) | |
| n | 3 |
| a | 2 |
| b | 1 |
| Initially we would check if n is equal to 0 or not which is not true, so we would return FIB(n-1,a+b,a) in the previous call, which is FIB(n,a,b) of the next recursive call or it would return (2,3,2). | |
| Next rec.call – Fib(2,3,2) – Fib(n,a,b) | |
| n | 2 |
| a | 3 |
| b | 2 |
| We would see if n is equal to 0 or not, which is not TRUE. So we would have the algorithm return FIB(n-1,a+b,a) in the current rec. call as the FIB(n,a,b) of the next rec. call.  a+b = 5  a= 3  Which would be equal to FIB (1,5,3) = FIB(n,a,b) of the recursive call. | |
| Next rec. call - FIB(1,5,3) = FIB(n,a,b) | |
| n | 1 |
| a | 5 |
| b | 3 |
| We would check if n would be equal to zero or not, which is not true. So we would return FIB(n-1,a+b,a) as the output of the current which would be input parameters of the next call.  As n is not equal to zero, the output would be equal to FIB(n-1,a+b,a) or that is equal to FIB(0,5+3,5) = FIB(0,8,5). | |
| Next rec. call – FIB(0,8,5) | |
| n | 0 |
| a | 8 |
| b | 5 |
| So the algorithm would check if n is 0 or not, which is true so it would return a according to :    So here a would be 8. | |
| **The final output would be equal to 8.** | |