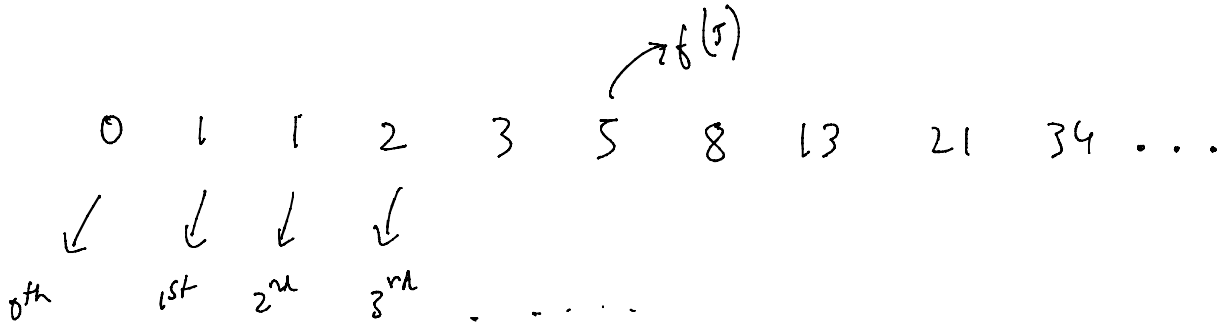
Multiple Recursion Calls :

```
f()
{
    f()
    f()   } 2 times
}
```

Fibonacci



N. ⟶ f(N) ⟶ N^{th} fibonacci nos.

f(3) ⟶ 2

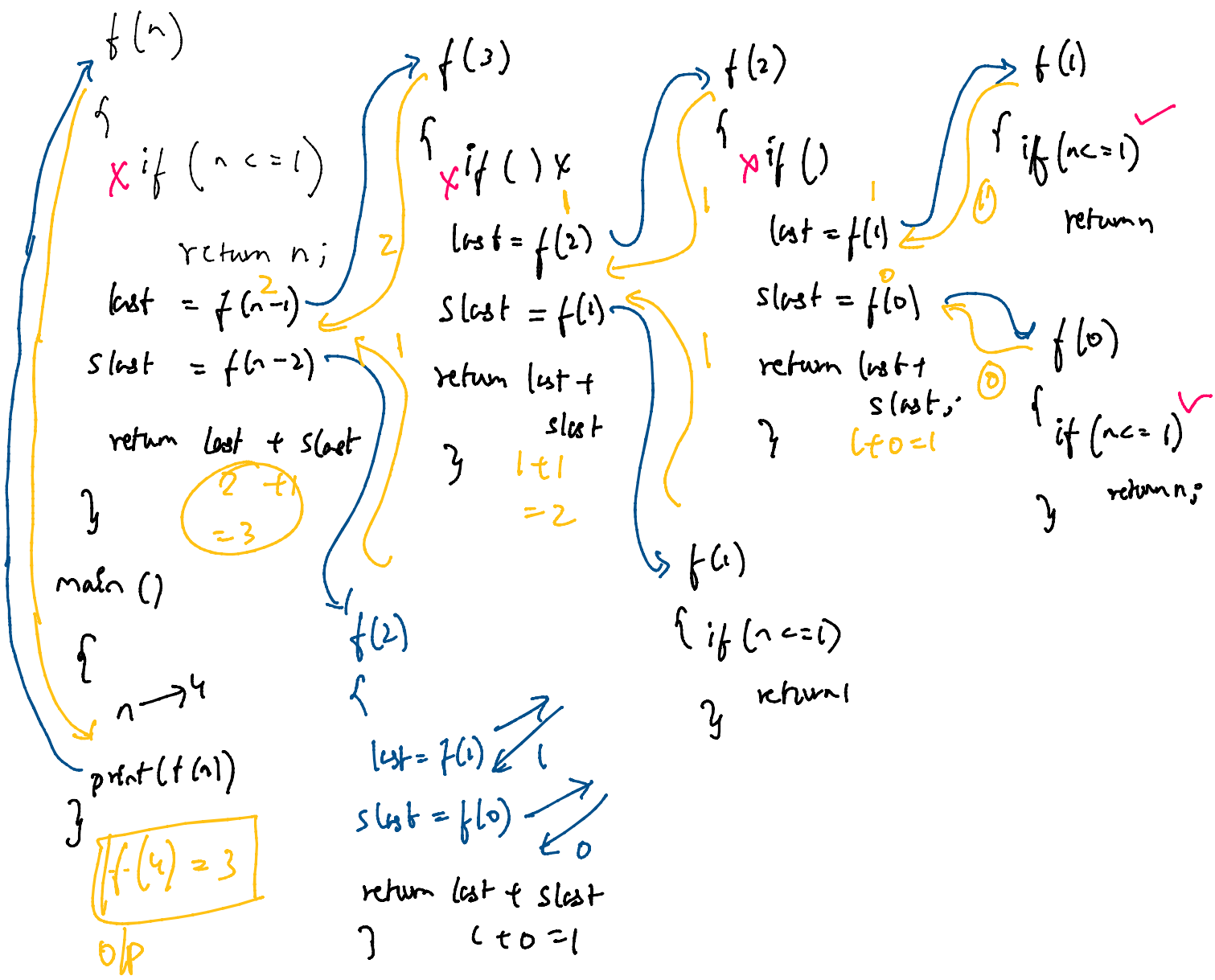
f(4) ⟶ 3

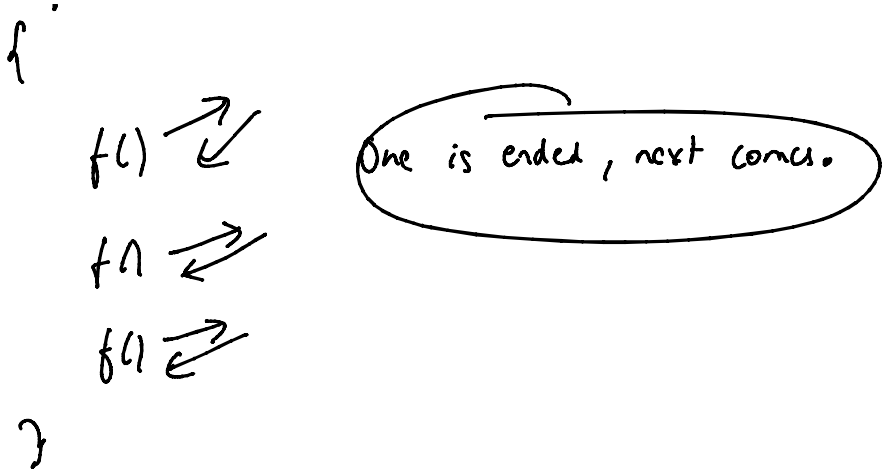
f(5) ⇒ f(4) + f(3)

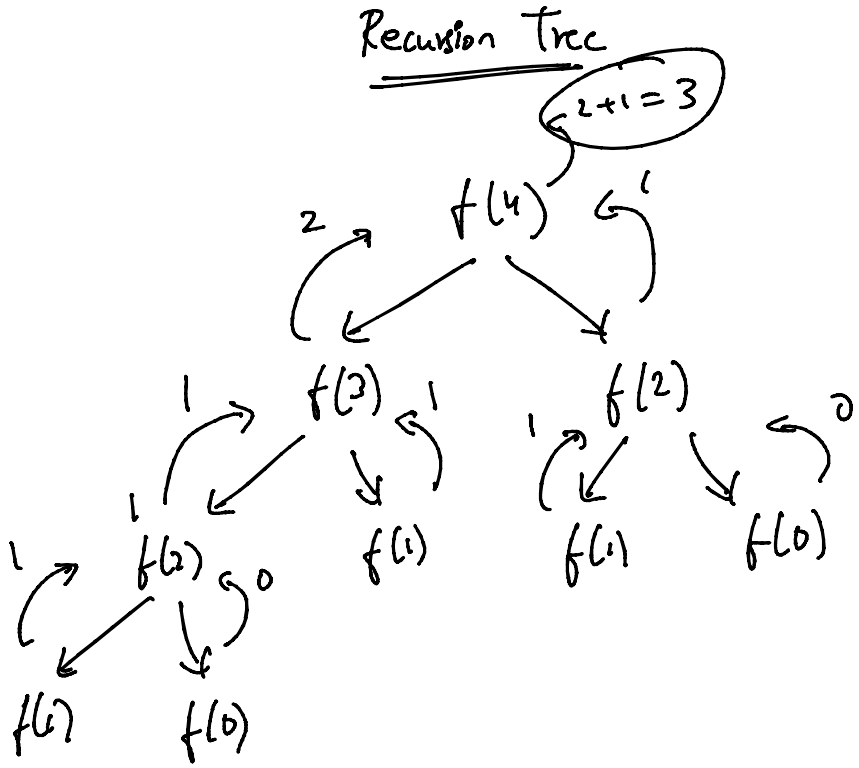
Iterative Approach

```
f[0] = 0      f[1] = 1
for (int i = 2 ⟶ N)
        f[i] = f(i-1) + f(i-2)
```

$$f(n) = f(n-1) + f(n-2)$$





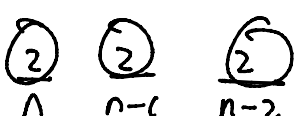


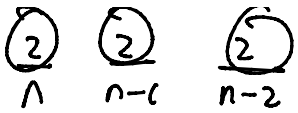

```cpp
#include <bits/stdc++.h>
using namespace std;
int fibo(int n){
    if ( n <= 1)
        return n;
    int last = fibo(n - 1);
    int secondLast = fibo(n - 2);
    return last + secondLast;
}

int main(){
    cout << fibo(4) << endl;
    return 0;
}
```

```
1
outputf.in
1 3
2
ished in 3.9s]
```

T.C ⇒ $O(2^n)$ Exponential in nature




2
2
2
1
n-1
n-2