

Αναδρομή

# Αναδρομή

- Τυπικό παράδειγμα αναδρομής

$$1! = 1$$

$$(n+1)! = (n+1) \times n!$$

- Υπάρχει τουλάχιστον μια βασική κατάσταση ή περίπτωση η οποία καθορίζει το αποτέλεσμα άμεσα για μια ειδική περίπτωση
- Υπάρχει τουλάχιστον μια αναδρομική περίπτωση η οποία προσδιορίζει το αποτέλεσμα με βάση την απάντηση σε μια πιο απλή περίπτωση του ίδιου προβλήματος

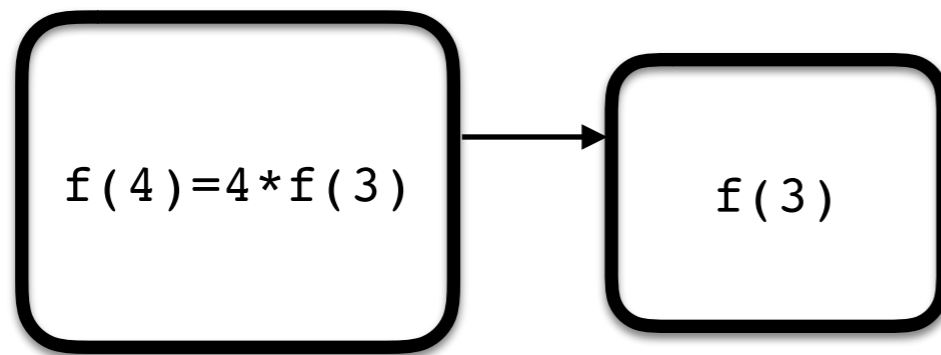
# Αναδρομή

Συνάρτηση - Παραγοντικό (αναδρομή)

```
def factorR(n):  
    if n==1:  
        return 1  
    else:  
        y=n*factorR(n-1)  
        return y
```

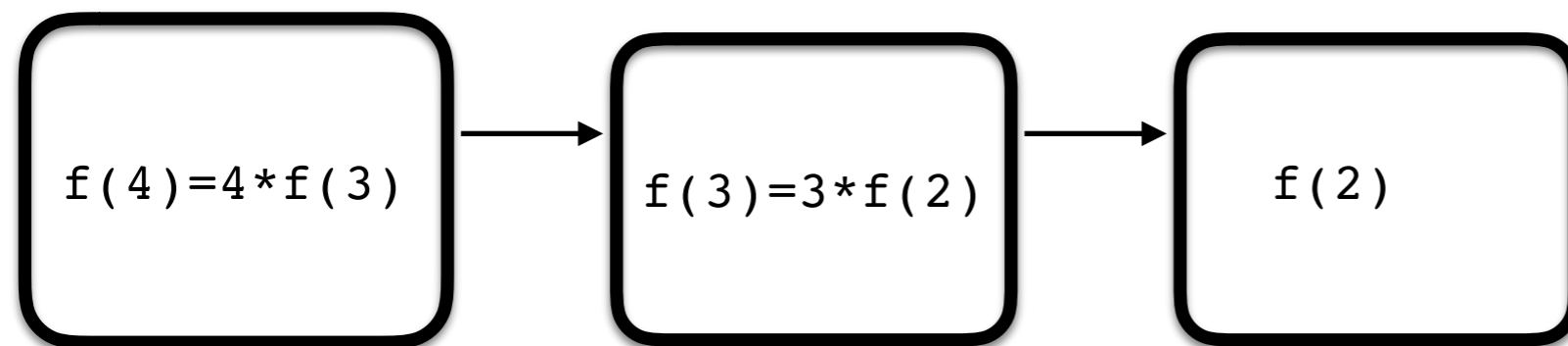
# Παραγοντικό

κλήση της  $f(4)$



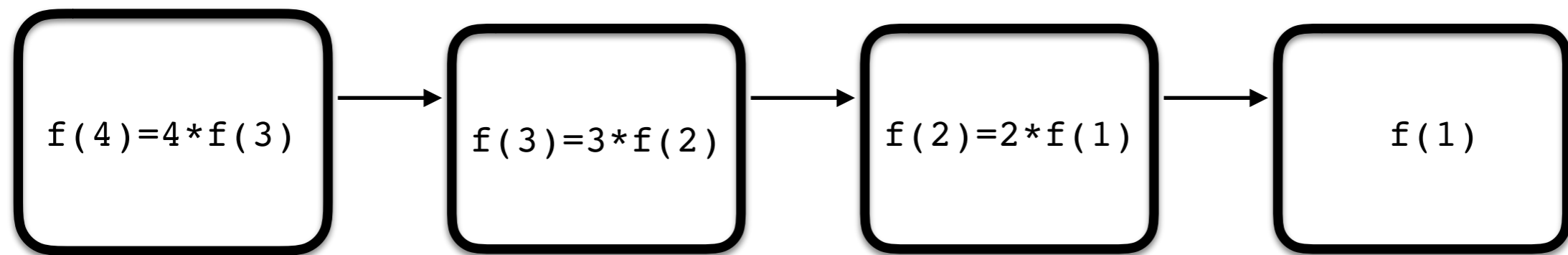
# Παραγοντικό

κλήση της  $f(4)$



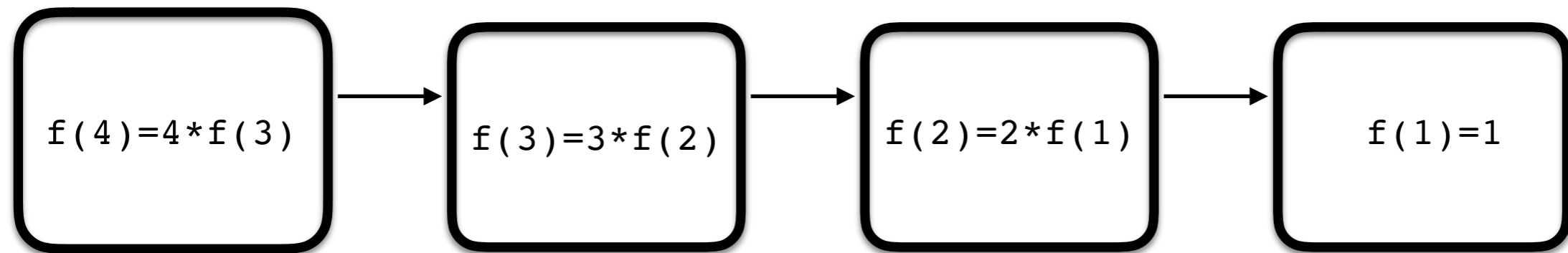
# Παραγοντικό

κλήση της  $f(4)$



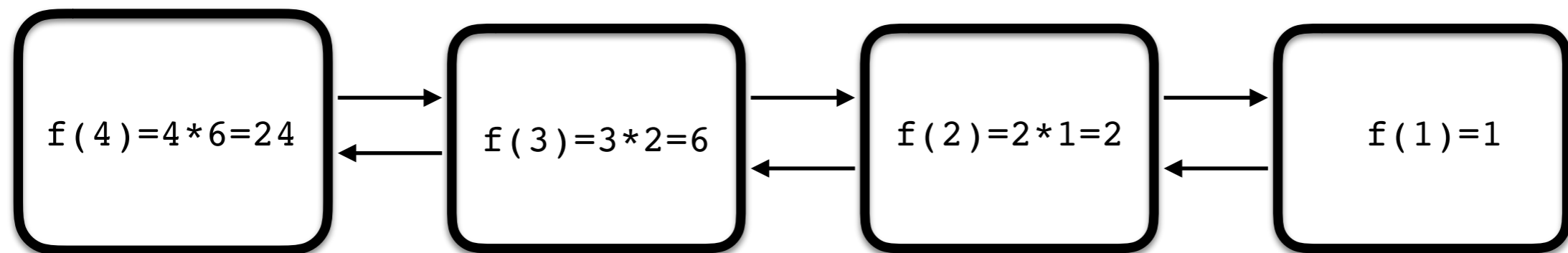
# Παραγοντικό

κλήση της  $f(4)$



# Παραγοντικό

κλήση της  $f(4)$





# Παραγοντικό

Επαναληπτική διαδικασία αντί αναδρομής

```
def factorI(n):  
    p=1  
    for i in range(2,n+1):  
        p=p*i  
    return p
```

# Fibonacci

$$F_0 = 0, \quad F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2}, \quad n \geq 2$$

```
def fib(n):  
    if n==0:  
        return 0  
    elif n==1:  
        return 1  
    else:  
        return fib(n-1)+fib(n-2)
```

# Fibonacci

$$f(4) = f(3) + f(2)$$



$$f(3) = f(2) + f(1)$$

# Fibonacci

$$f(4) = f(3) + f(2)$$



$$f(3) = f(2) + f(1)$$



$$f(2) = f(1) + f(0)$$

# Fibonacci

$$f(4) = f(3) + f(2)$$



$$f(3) = f(2) + f(1)$$

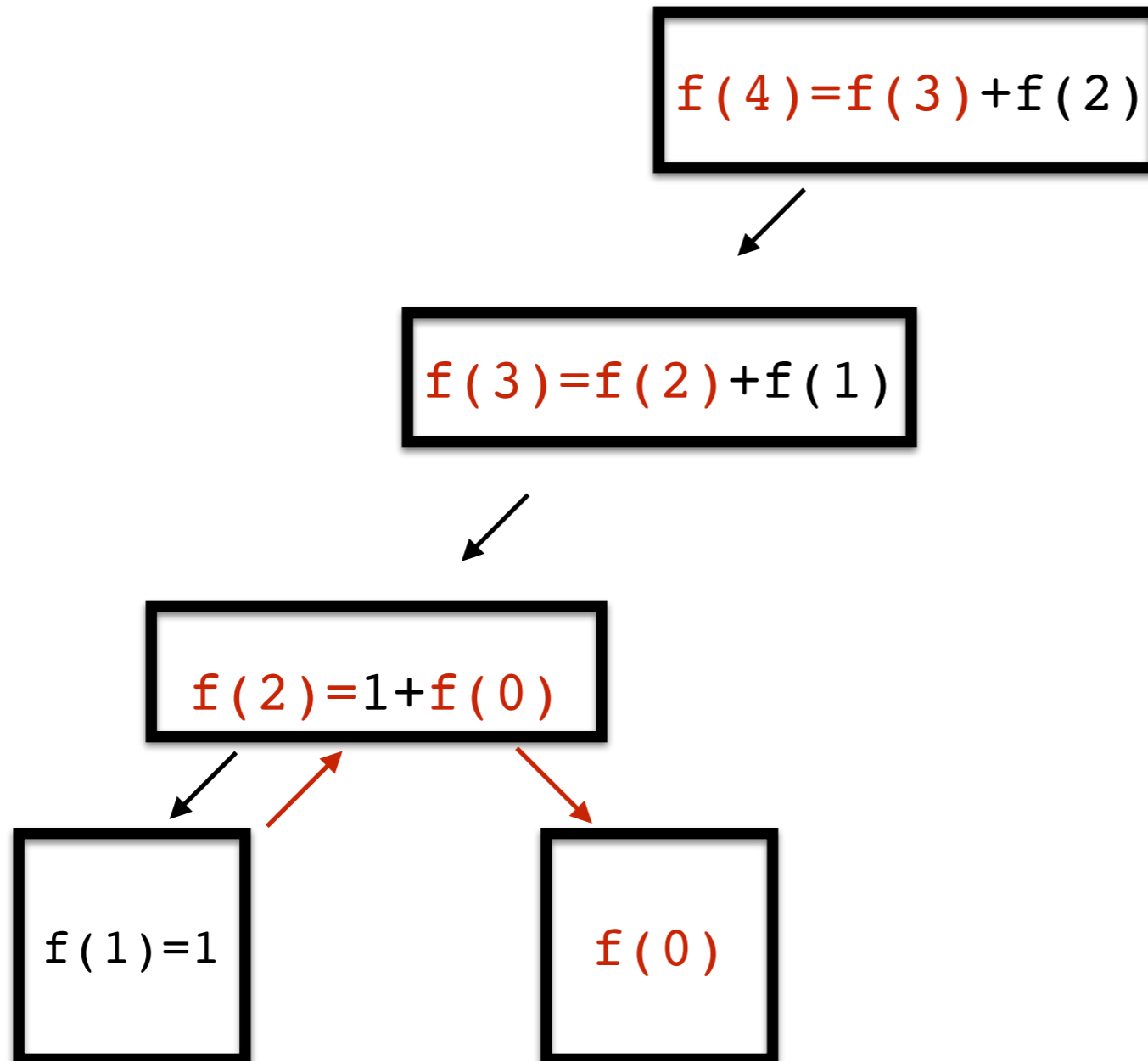


$$f(2) = f(1) + f(0)$$

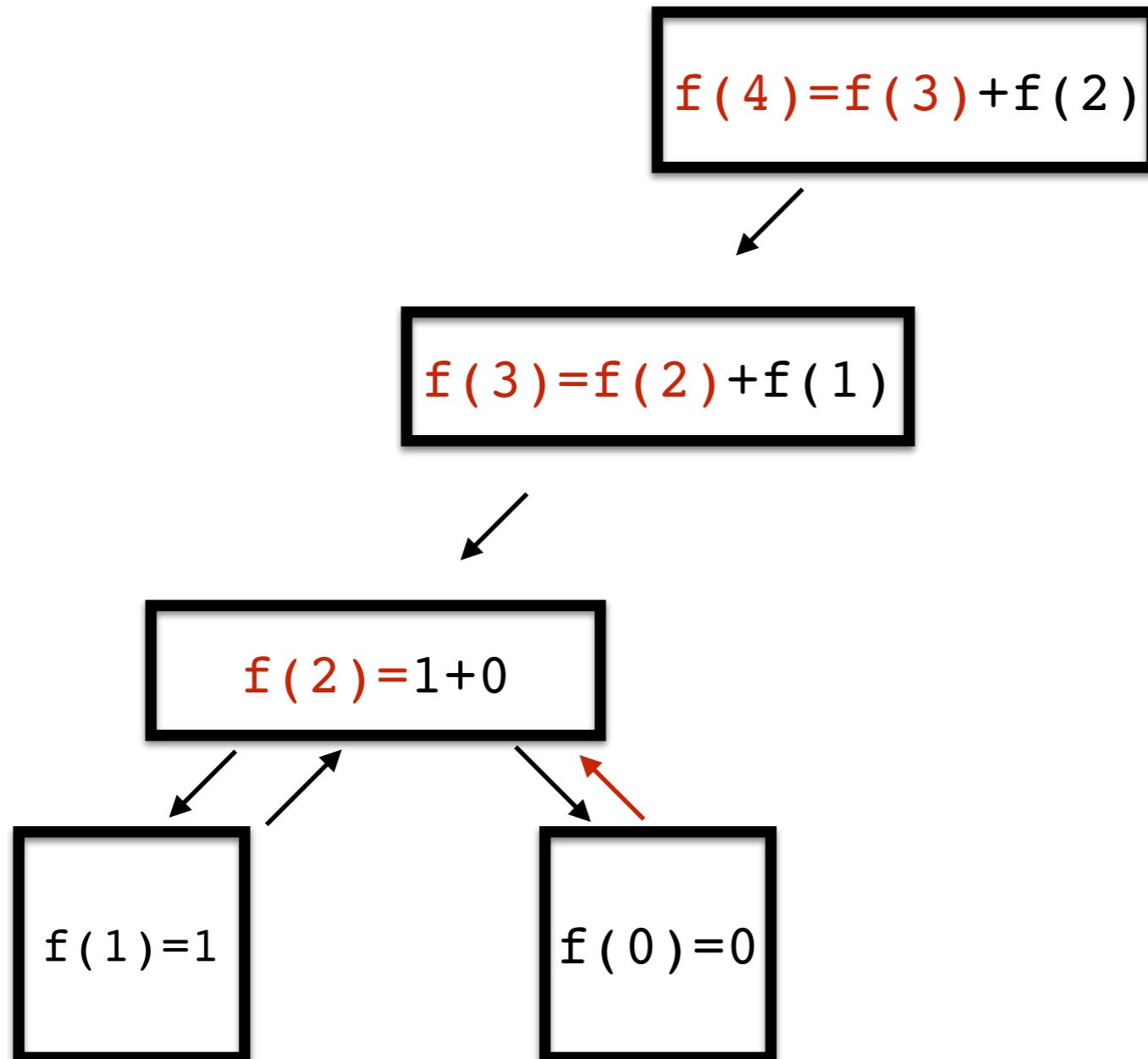


$$f(1)$$

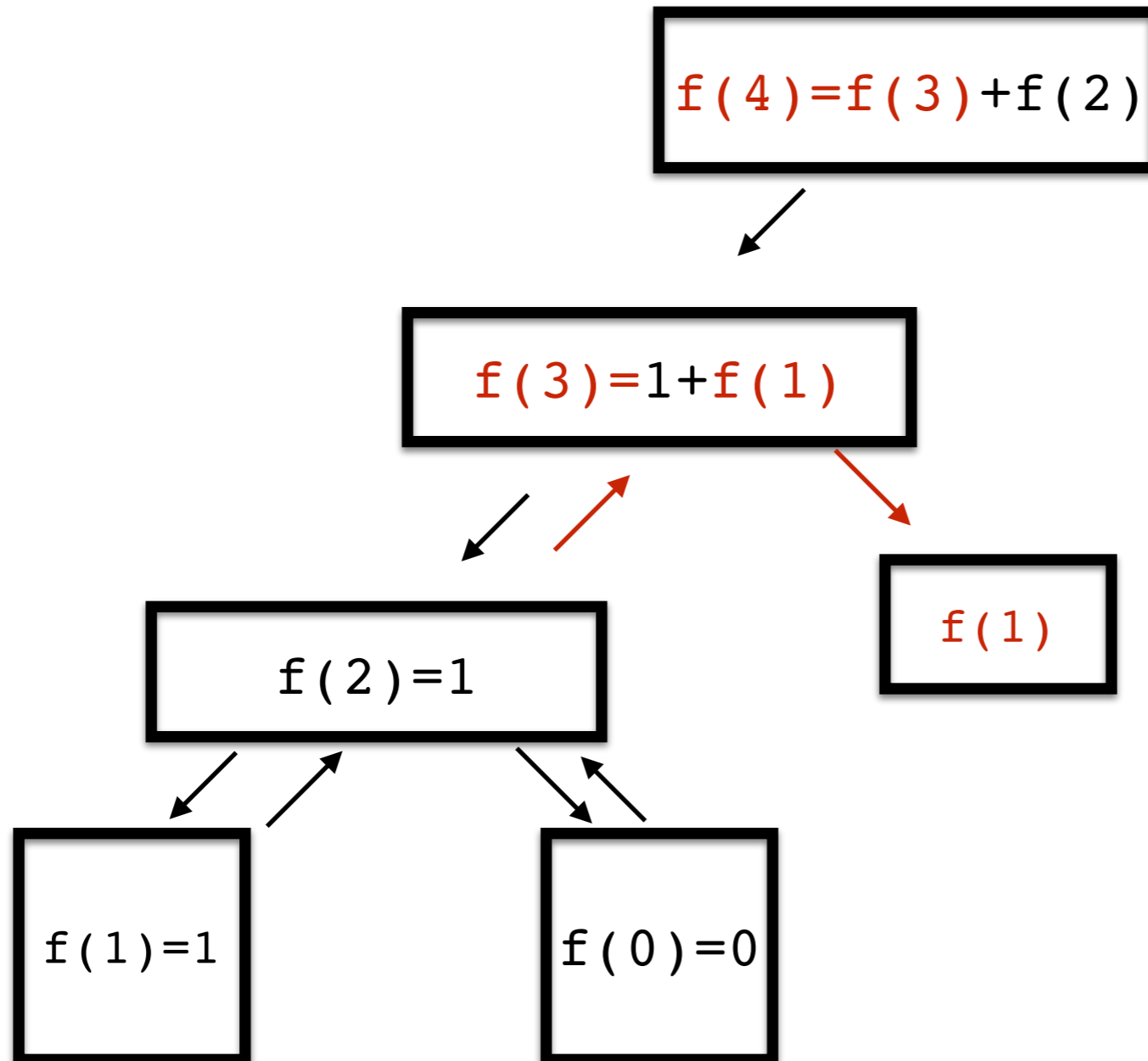
# Fibonacci



# Fibonacci

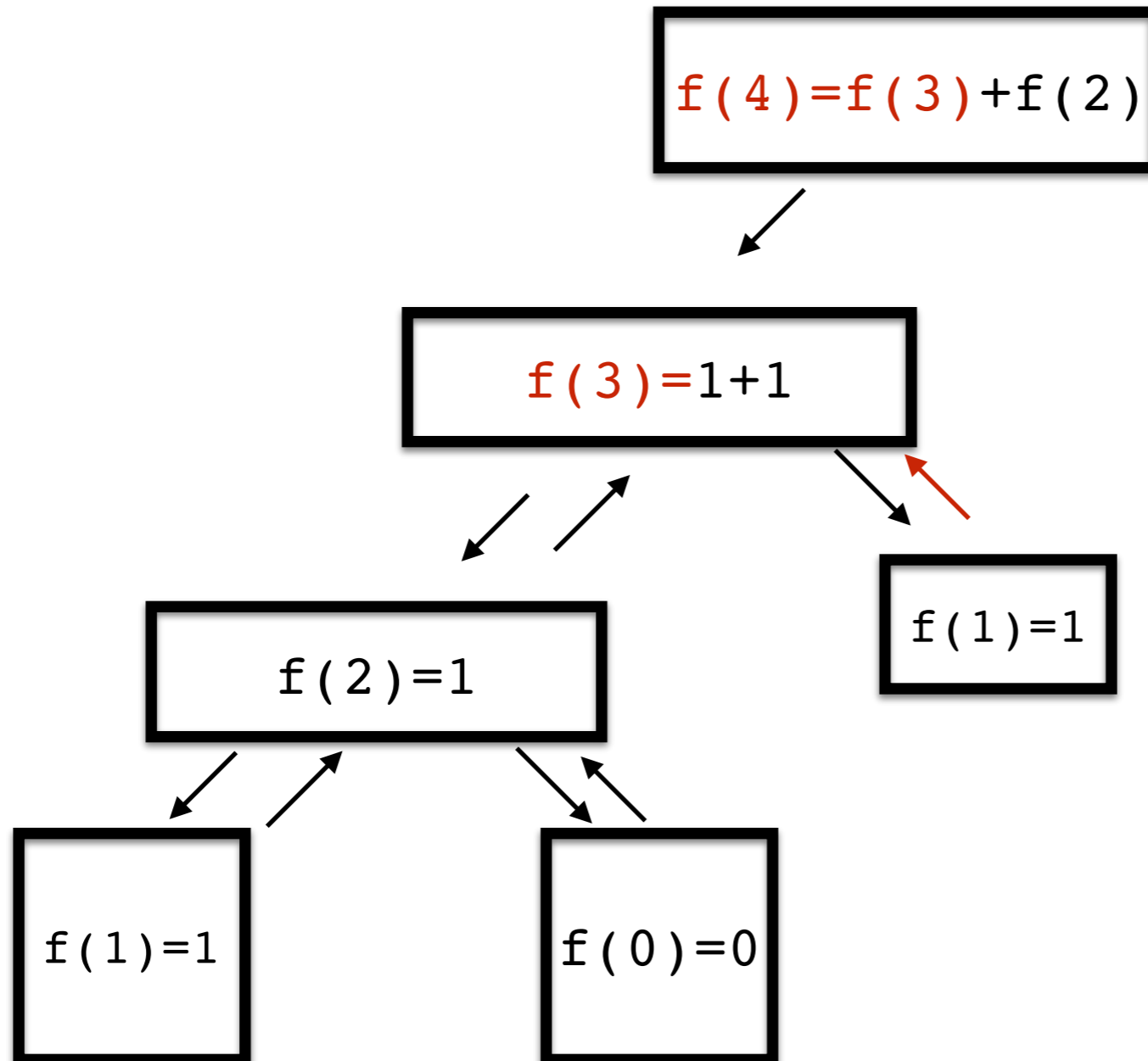


# Fibonacci

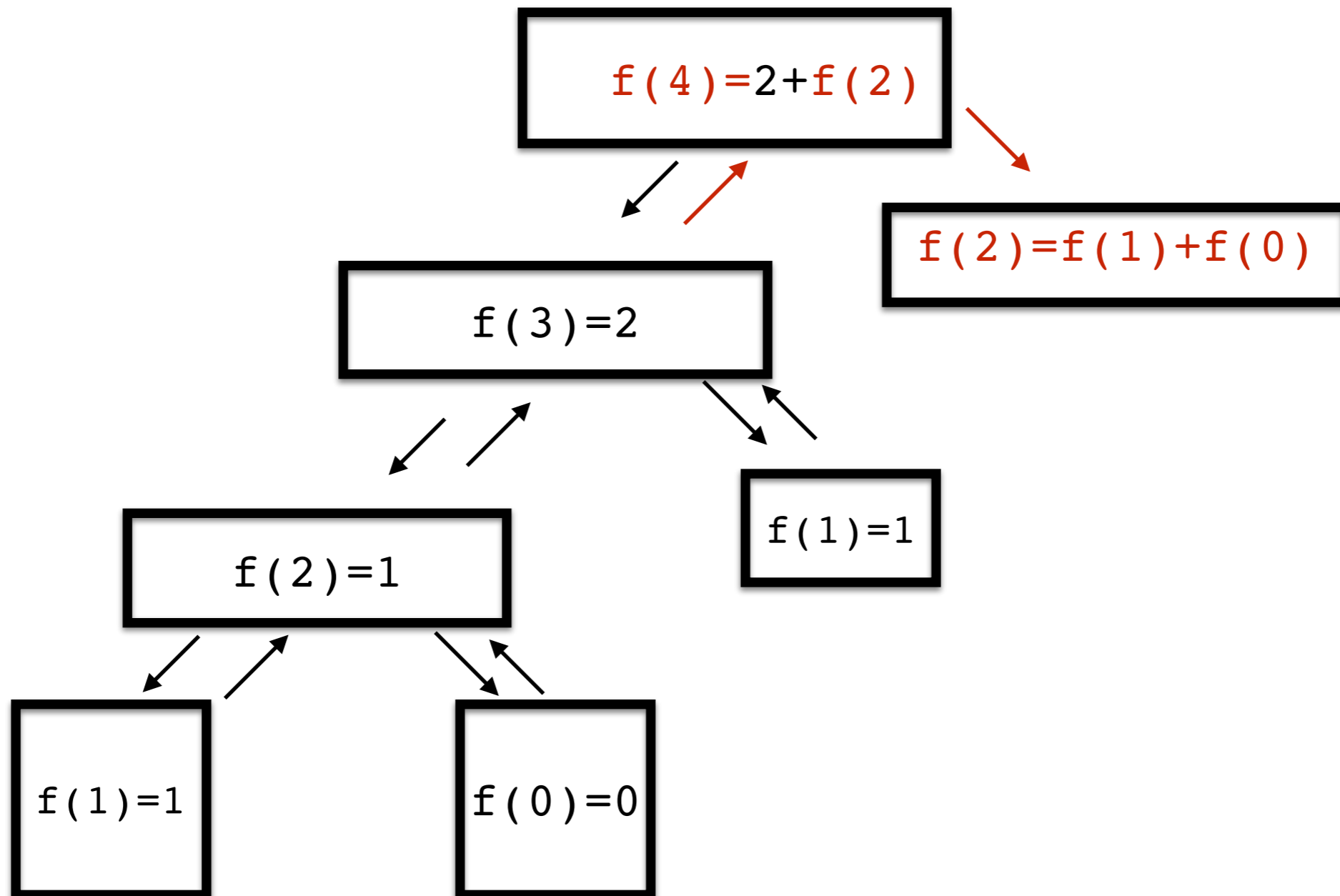




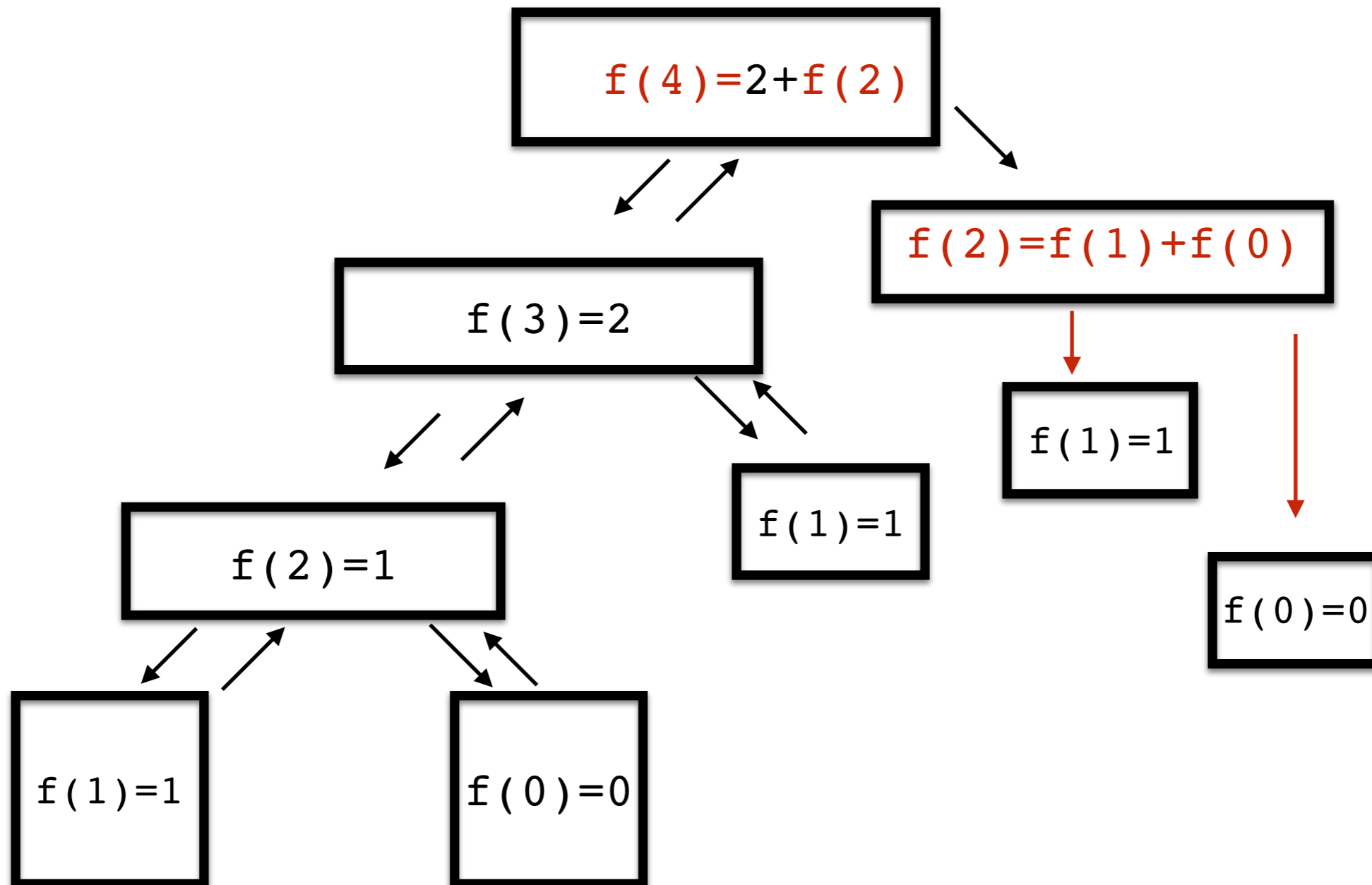
# Fibonacci



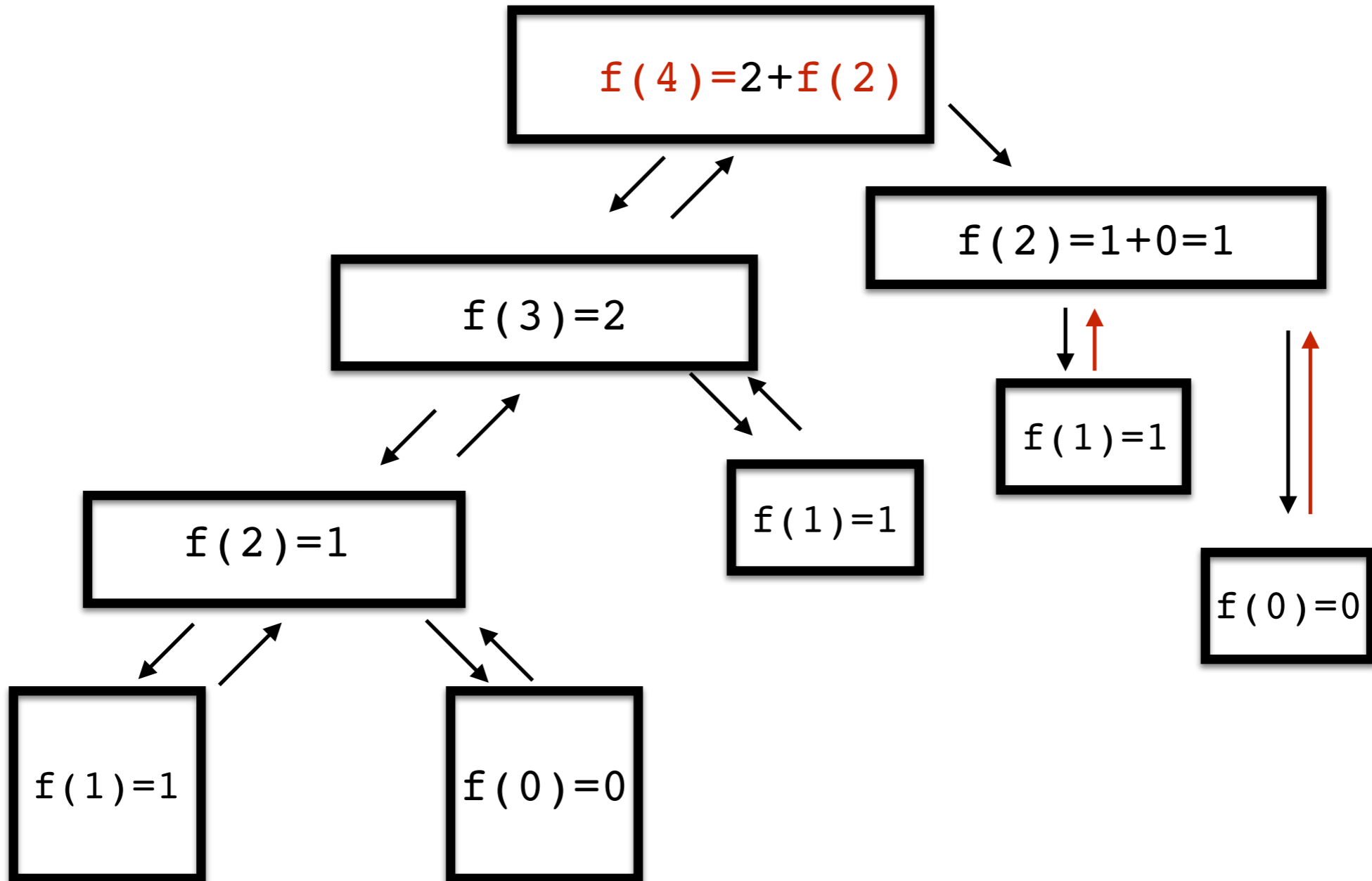
# Fibonacci



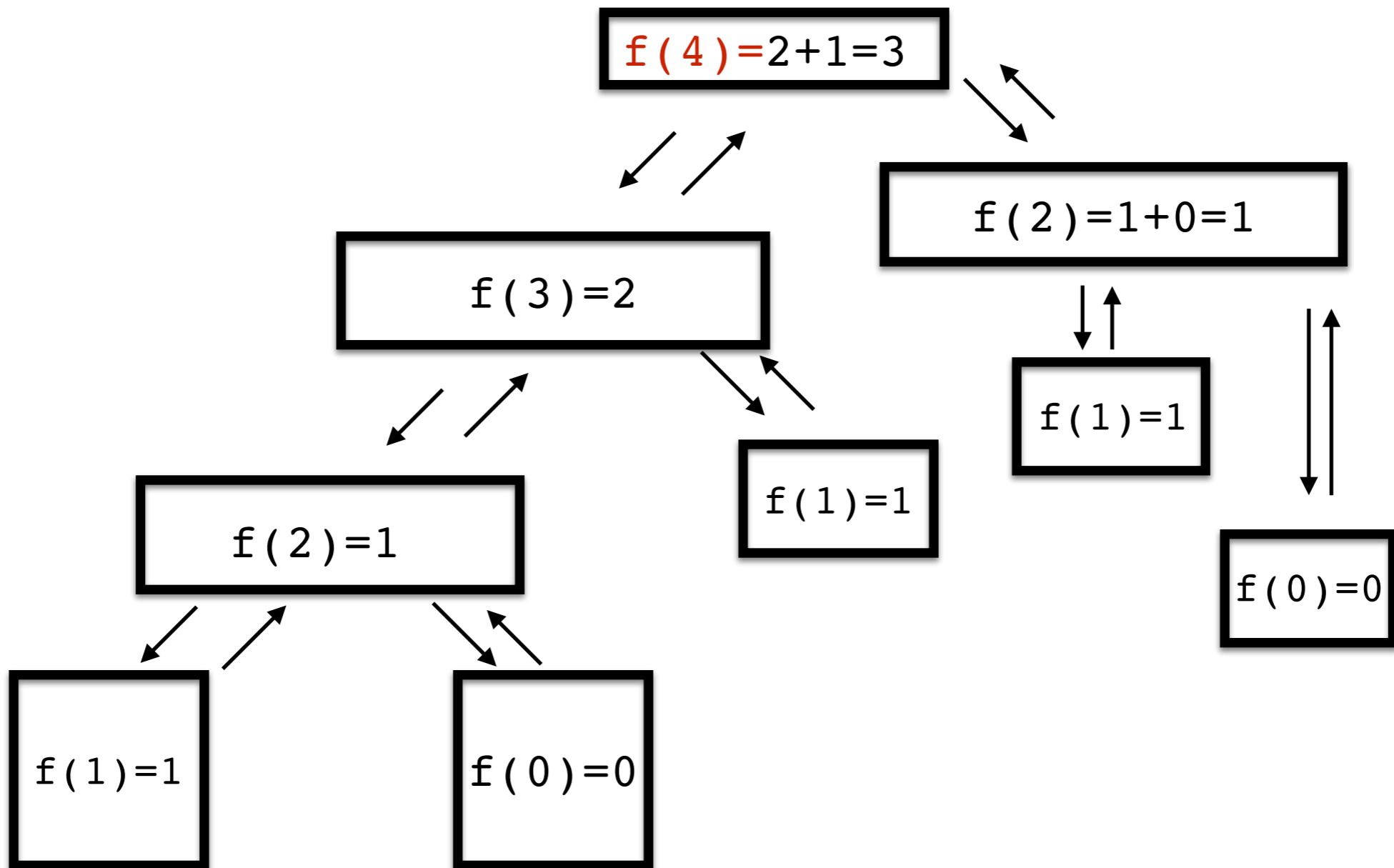
# Fibonacci



# Fibonacci



# Fibonacci

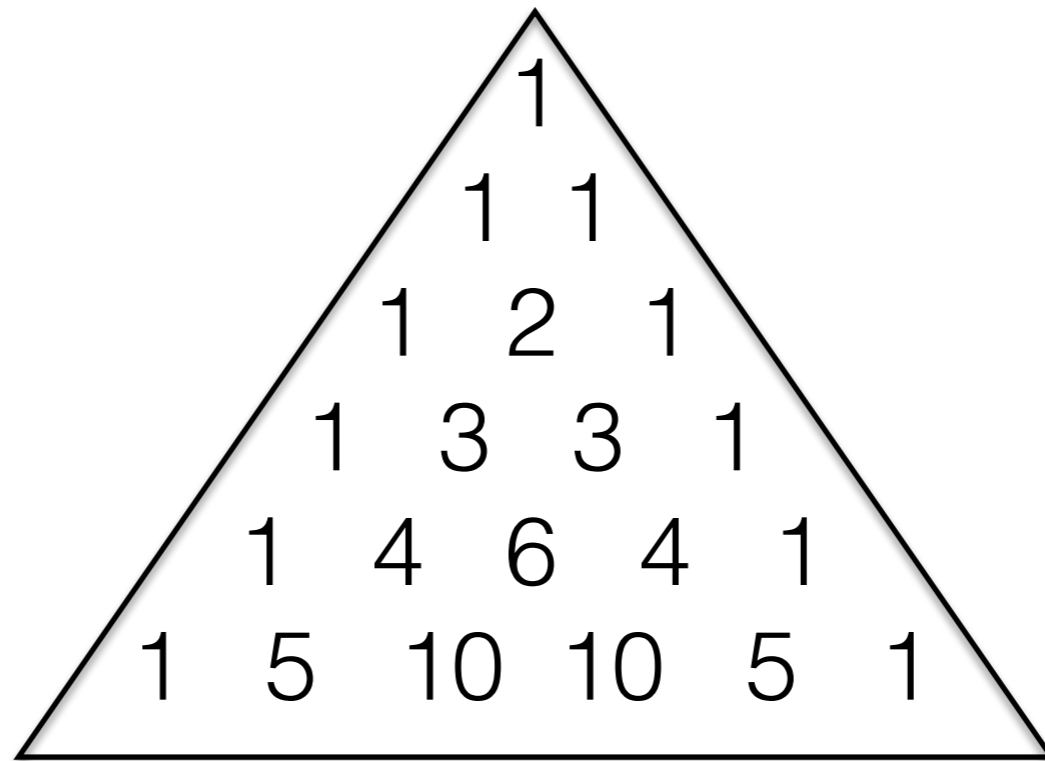


# Πολλαπλασιασμός

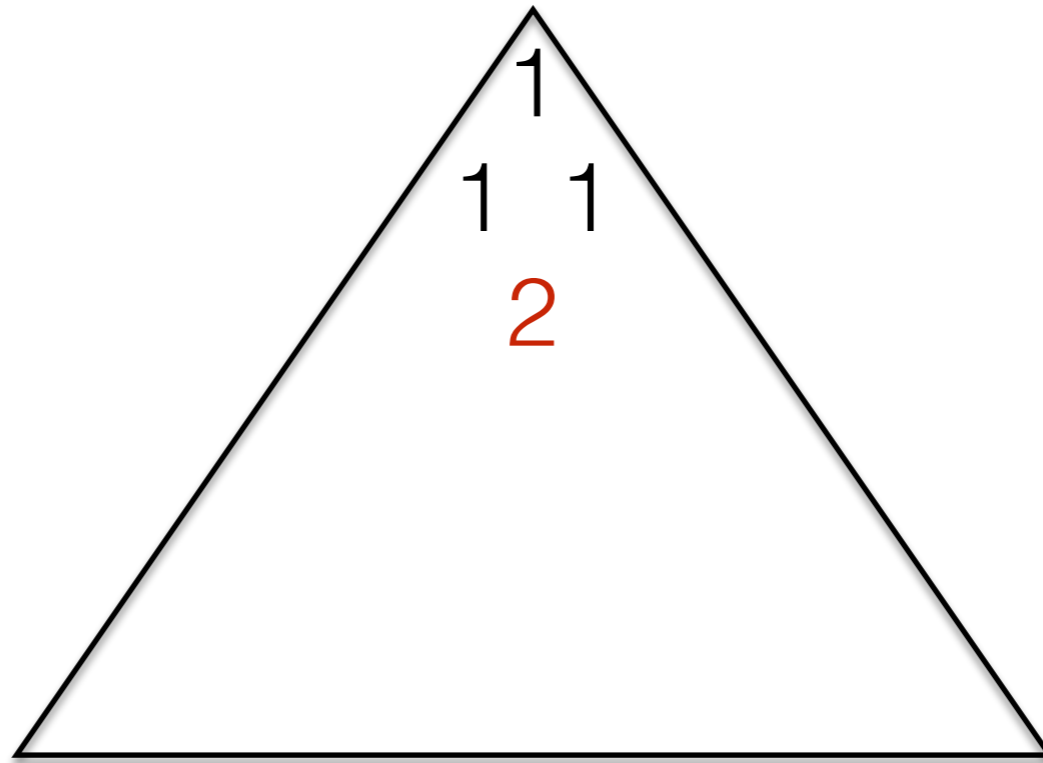
- $3 * n = 3 + 3 + 3 + 3 + \dots + 3$  (n φορές)
- $n = 1 \longrightarrow 3$  (βασική κατάσταση)
- $n \longrightarrow 3 + (3 * (n - 1))$  (Αναδρομική σχέση)

```
def mult3(n):  
    if n==1:  
        return 3  
    else:  
        return mult3(n-1)+3
```

# Τρίγωνο Pascal

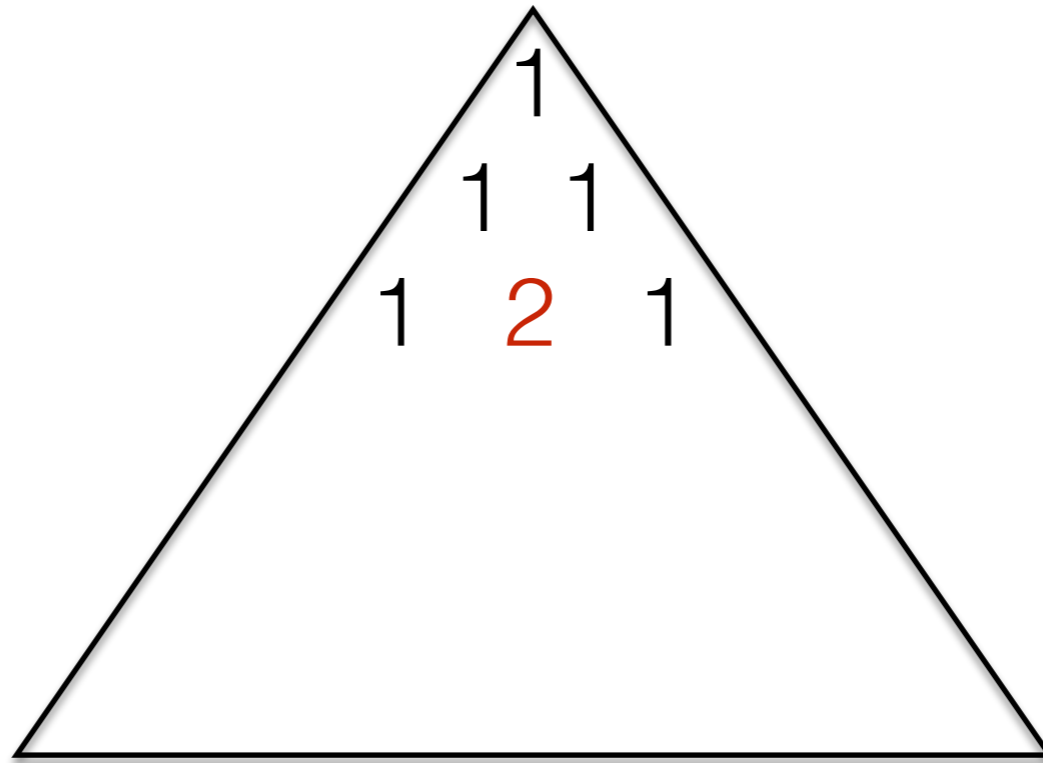


# Τρίγωνο Pascal

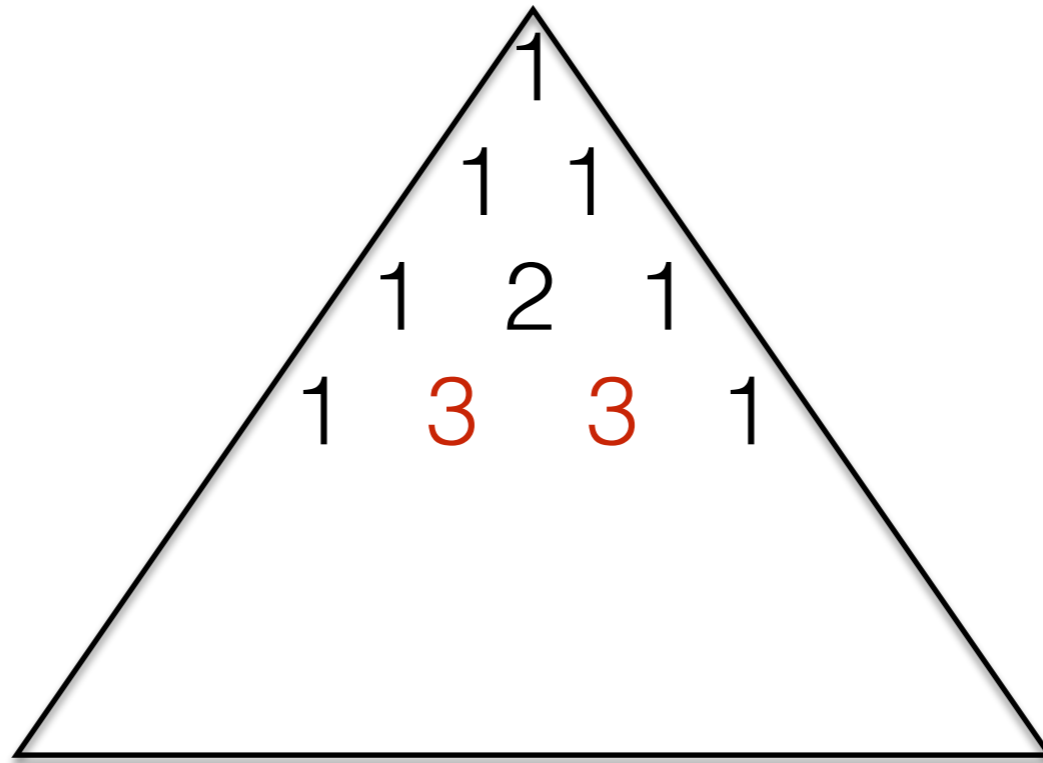




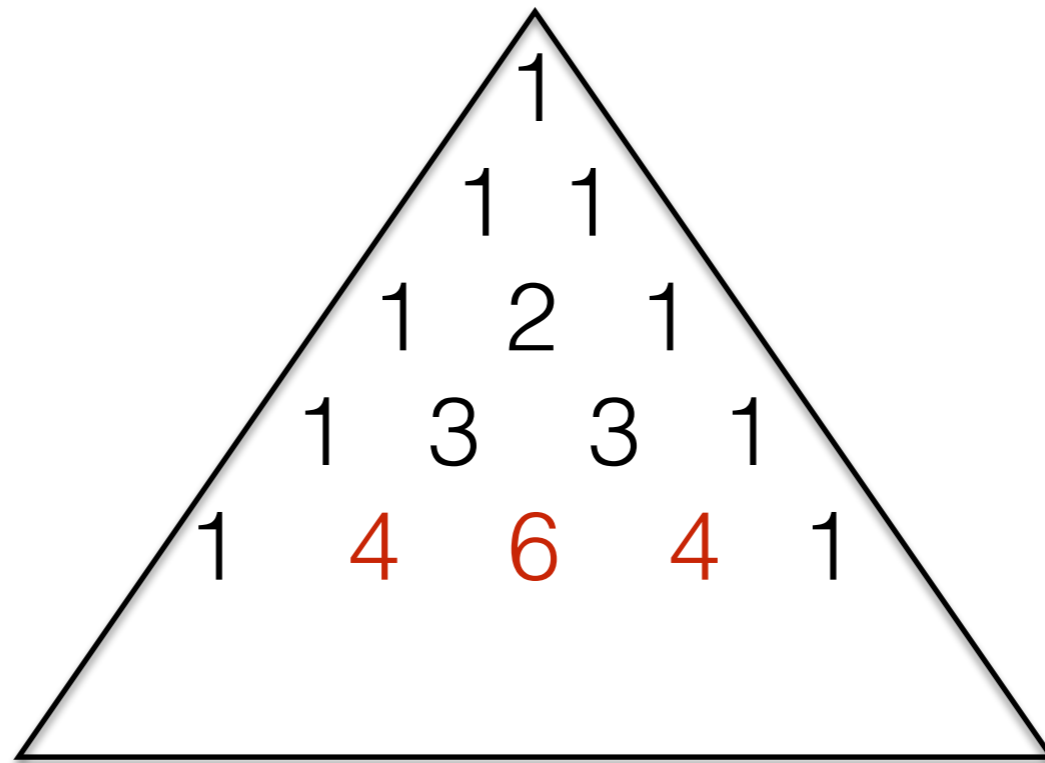
# Τρίγωνο Pascal



# Τρίγωνο Pascal



# Τρίγωνο Pascal



# Τρίγωνο Pascal

- Αναδρομική σχέση: Η  $n$ -ιοστή γραμμή βασίζεται στην  $(n-1)$ -ιοστή γραμμή
- Η 1η και η 2η δεύτερη γραμμή είναι δοσμένες

# Τρίγωνο Pascal

$n$ -ιοστή γραμμή

- $n$  στοιχεία
- Το πρώτο και τελευταίο είναι 1
- Οι υπόλοιποι  $n-2$  όροι προκύπτουν ως εξής:  
Για  $i$  από 2 έως  $n-1$   
Ο  $i$  όρος είναι το άθροισμα του  $i$  και  $i-1$  όρου της προηγούμενης γραμμής

Βασική κατάσταση

- είναι οι 2 πρώτες γραμμές

# Τρίγωνο Pascal

## Υλοποίηση

- Κάθε γραμμή είναι μια λίστα.
- Η αναδρομική συνάρτηση που θα κατασκευάσουμε θα επιστρέφει τη  $n$ -οστή γραμμή του τριγώνου Pascal

# Τρίγωνο Pascal

```
def pascal(n):  
    if n==1:  
        return [1]  
    else:  
        line=[1]  
        p_line=pascal(n-1)  
        l=len(p_line)  
        for i in range(l-1):  
            line.append(p_line[i]+p_line[i+1])  
        line.append(1)  
        return line  
  
print(pascal(6))
```