Fibonacci

0, 1, 1, 2, 3, 5, 8, 13, ...

$$F_n = \begin{cases} F_{n-1} + F_{n-2}, & \text{αν } n > 1 \\ 1, & \text{αν } n = 1 \\ 0, & \text{αν } n = 0 \end{cases}$$

→ $f_n > 2^n$

Χρόνος ως προς. με είσοδο $n \in \mathbb{N}$ (# βημάτων)

Συνάρτηση fib1(n):

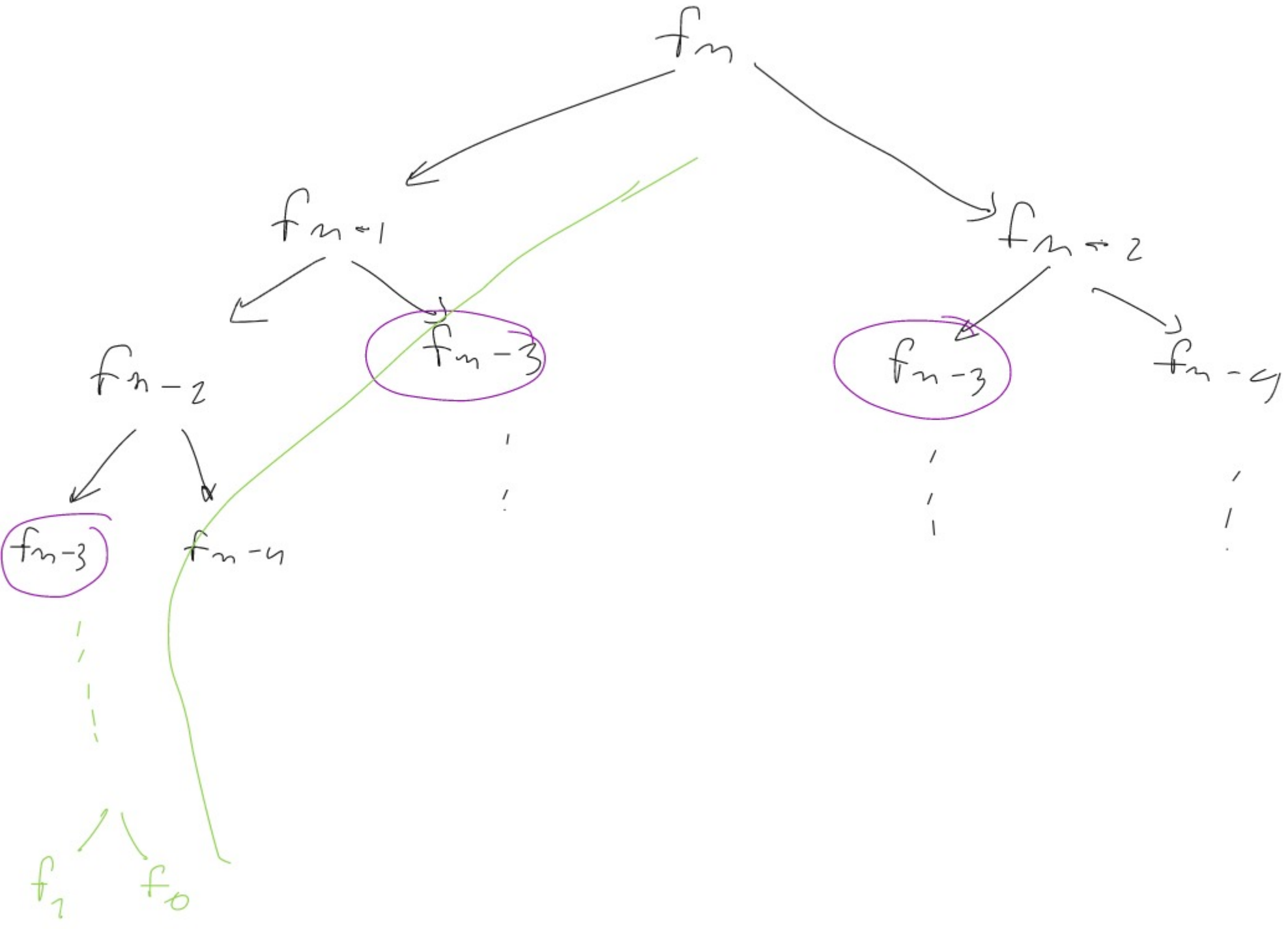
```
if   n=0
     return  0

if   n=1
     return  1

else
     return  fib1(n-1) + fib1(n-2)
```

Λύσιμο:

$$\begin{cases} T(n) = 3 + T(n-1) + T(n-2) \\ T(1) = 2 \\ T(0) = 1 \end{cases}$$

$n = 200$ ο χρόνος για fib1(200) είναι $> 2^{92}$ sec

Diese Aufgabe visiert:



Εναλλακτικά fib2(n) :

→ 10000000

→ ♡

if n=0 return 0

Δημιουργία μια ακολουθίας f[0 ... n]

f[0] = 0 , f[1] = 1

for i = 2, ..., n

f[i] = f[i-1] + f[i-2]

return f[n]

Ανάλυση:

$$T(n) = 1 + 1 + 2 + (n-1)\cdot 2 = \underbrace{c_1 \cdot n + c_2}_{\text{1η}} = O(n)$$

5/15

Αν η ακολουθία έχει n-στοιχεία : $n \cdot f_n$

Ασυμπτωτική Ανάλυση (ασυμπτωτικός συμβολισμός):

Ορισμός: $f, g : \mathbb{N} \to \mathbb{R}^+$. Λέμε ότι $f = O(g)$ αν υπάρχει $c > 0$ τ.ώ. να υπάρχει n_0 τέτοιο ώστε για κάθε $n \geq n_0$ να ισχύει $f(n) \leq c \cdot g(n)$.

($c > 0$ και n_0 τ.ώ. για κάθε $n \geq n_0$ $f(n) \leq c \cdot g(n)$)

$f(n) = 2n + 20$

$g(n) = n^2$

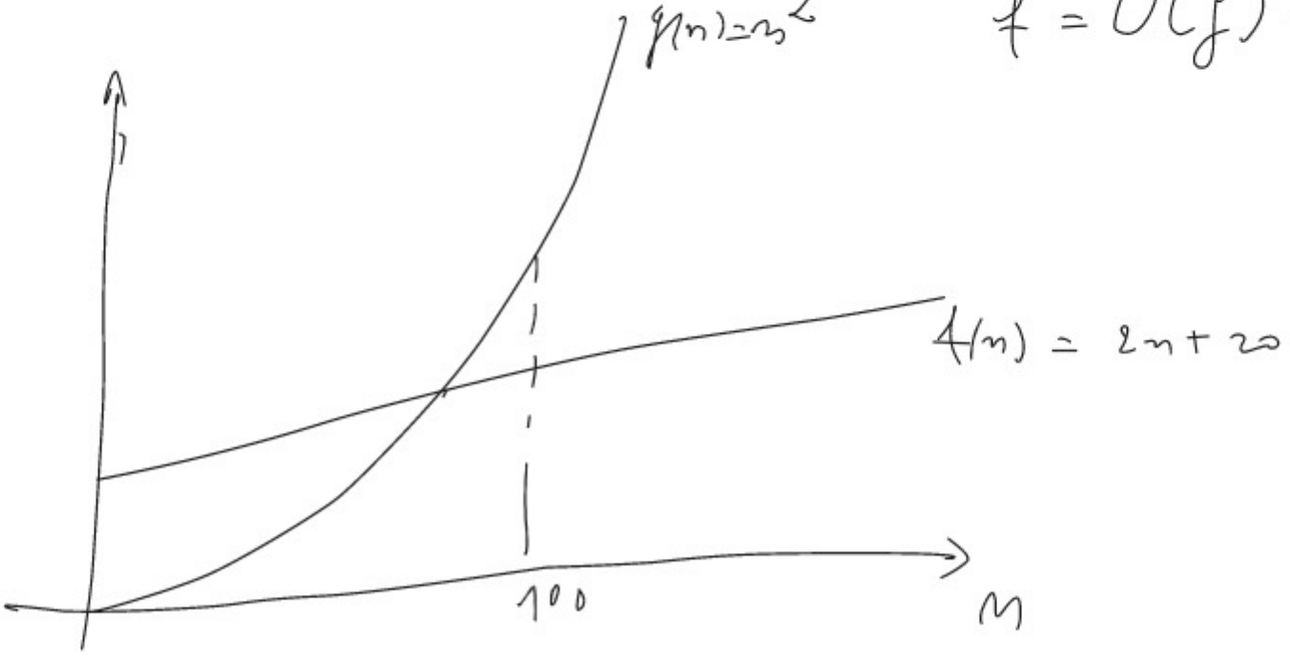
$n_0 = 100$

$c = 1$

για $n \geq 100$ $f(n) \leq 1 \cdot g(n)$

$\Downarrow$

$f = O(g)$.



$f(n) = 2n + 20 = f_1(n)$

$(g(n) = n^2 = f_2(n))$

$h(n) = n + 10 = f_3(n)$

$h = O(f)$

$f = O(h)$ n_0 $c = 2$

$n_0 = 100$

Finding:

- $f = \Omega(g)$ then $g = O(f)$

  (n.x. $f_2 = \Omega(f_1)$)

- $f = \Theta(g)$ then $f = O(g)$ (n.x. $f_1 = \Theta(f_3)$)

  $g = O(f)$

15. $n^4 + 3n^3 + 1 = O(n^4)$

1. Αφαίρεση των σταθερών $3n^2 = O(n^2)$

π.χ. $3n^2 \rightsquigarrow n^2$

2. Το n^a επικρατεί του n^b με $a > b$

$15n^3 + 2n^2 = O(n^3)$

π.χ. n^3 επικρατεί του n

3. Εκθετικοί όροι επικρατούν των πολυωνυμικών

$2^n + n^{15} + 3 = O(2^n)$

π.χ. 2^n επικρατεί του n^{15}

4. Πολλαπλοί όροι επικρατούν των λογαριθμικών

$(\log n)^3 = O(n)$

π.χ. n επικρατεί του $n \log n$

$n \log n = O(n^2)$