$nphir(i=1)$

if $i == 10$:
return i

$\Rightarrow 1 + 1/nphir(2)$

$nphir(2) \Rightarrow 1 + 2/nphir(3)$

$nphir(3) \Rightarrow 1 + 3/nphir(4)$

$nphir(4) \Rightarrow 1 + 4/nphir(5)$

$nphir(5) \Rightarrow 1 + 5/nphir(6)$

$nphir(6) \Rightarrow 1 + 6/\cancel{nphir(7)}$ $\quad 1 + 6/1 + 7/1 + 8/1 + \frac{9}{10}$

$nphir(7) \Rightarrow 1 + 7/\cancel{nphir(8)}$ $\quad 1 + 7/1 + \frac{8}{1 + \frac{9}{10}}$

$nphir(8) \Rightarrow 1 + 8/\cancel{nphir(9)}$ $\quad 1 + \frac{8}{1 + \frac{9}{10}}$

$nphir(9) \Rightarrow 1 + 9/\cancel{nphir(10)}$ $\quad 1 + \frac{9}{10}$

$nphir(10) \Rightarrow 10$

$f_1 = 1$

$f_2 = 1$

$f_3 = 2$

$f_4 = 3$

$f_5 = 5$

$f_6 = 8 = f_5 + f_4$

$f_7 = 13 = f_6 + f_5$

$\frac{f_i}{f_{i+1}} \sim \varphi \quad i \to \infty$

$h=0, \ k=1 \Rightarrow 1$

$h=1, \ k=1 \Rightarrow 1$

$h=1, \ k=2 \Rightarrow 2$

$h=2, \ k=3 \Rightarrow 1.5$

Complejidad Computacional

El análisis del tiempo y el espacio que ocupa un <u>algoritmo</u>.

1. Localizar el menor valor de una lista de n valores.

Idea: Usar "<" 2^{n-1}

$[a_0, a_1, a_2, a_3, \ldots]$

(1) $a_0 < a_1$? $\xrightarrow{si}$ $a_0 < a_2$? $\xrightarrow{si}$ $a_0 < a_3$? $\xrightarrow{si}$ ....

$a_0 < a_2$? $\xrightarrow{no}$

$a_0 < a_3$? $\xrightarrow{no}$

$a_0 < a_1$? $\xrightarrow{no}$ $a_1 < a_2$? $\xrightarrow{si}$

$a_1 < a_2$? $\xrightarrow{si}$

(2) $a_0 < a_1$? $\xrightarrow{si}$ $min = a_0$

$a_0 < a_1$? $\xrightarrow{no}$ $min = a_1$

$min < a_2$? $\xrightarrow{si}$ —

$min < a_2$? $\xrightarrow{no}$ $min = a_2$

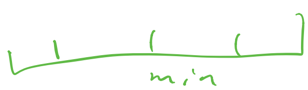
$min < a_3$? $\xrightarrow{si}$ —

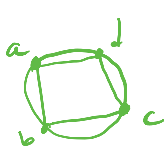
$min < a_3$? $\xrightarrow{no}$ $min = a_3$

→ min

n	2^n
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048

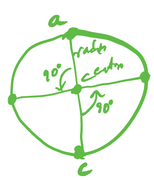
n	n-1
0	-1
1	0
2	1
3	2
4	3

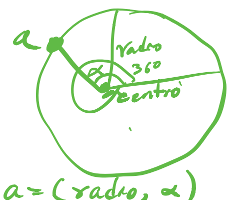




Tareas pendientes

ensayo
dibujos
trascendentes
(iterativo a recursivo)





a = (radio, α)

$b = (\text{radio}, \alpha + 90°)$

$c = (\text{radio}, \alpha + 180°)$

$d = (\text{radio}, \alpha + 270°)$