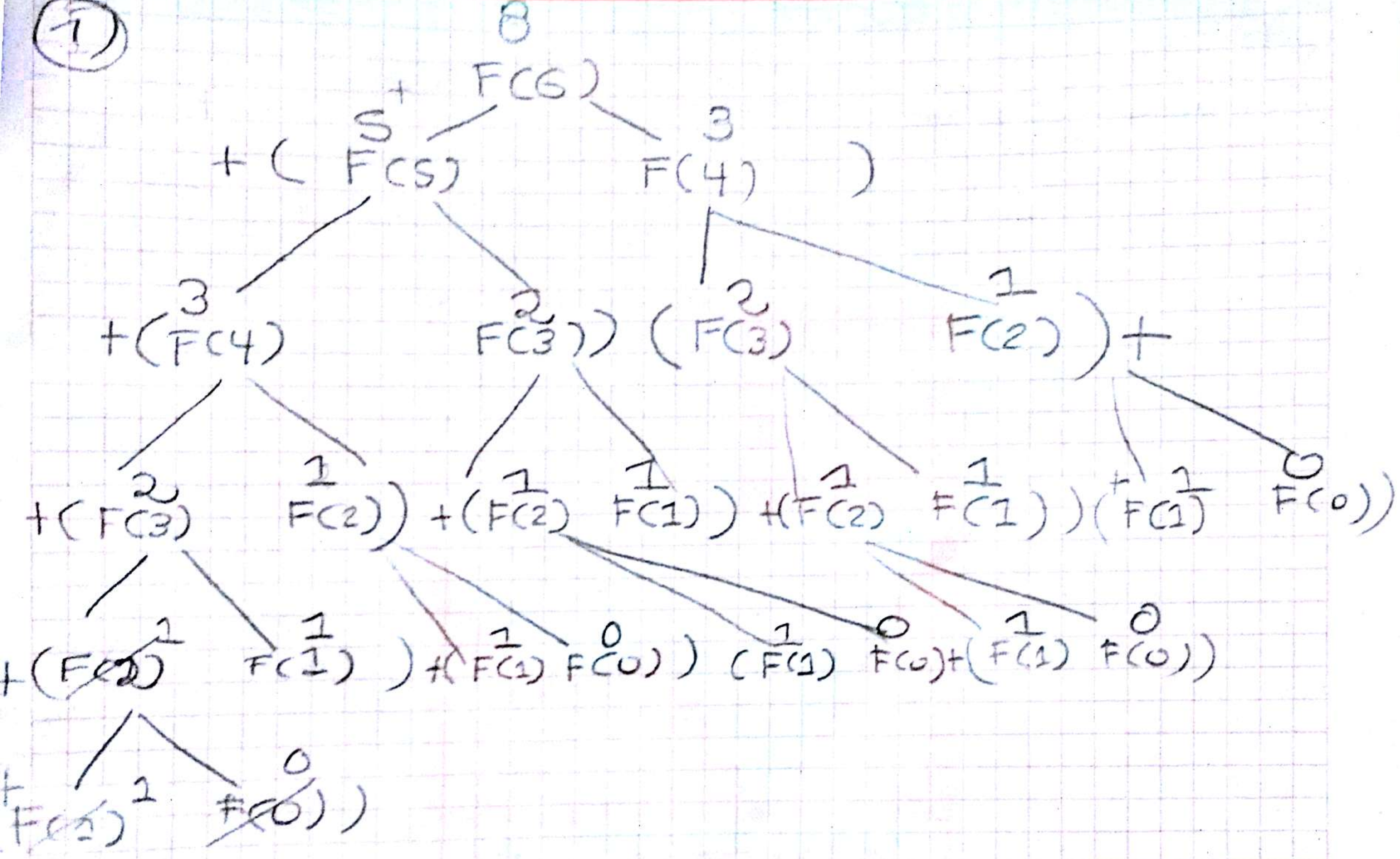
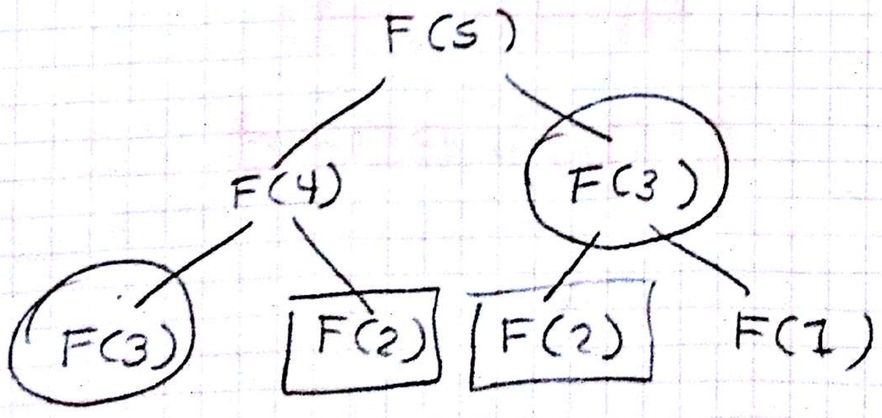
1


8
F(6)
+(F(5) F(4))
+(F(4) F(3)) (F(3) F(2))+
+(F(3) F(2)) +(F(2) F(1)) +(F(2) F(1)) (F(1) F(0))
+(F(2) F(1)) +(F(1) F(0)) (F(1) F(0)+(F(1) F(0))
F(1) F(0))

Ejemplo:

2


F(5)
F(4)
F(3)
F(3)
F(2)
F(2)
F(1)

③ Caracteriza

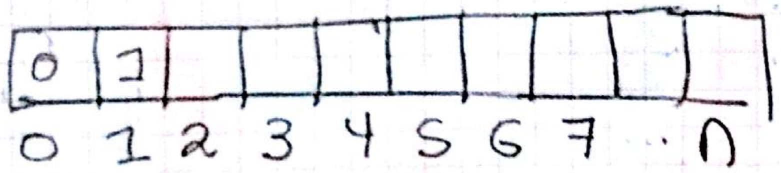
$$\boxed{F(6) = F(5) + F(4)}$$

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

Decisión unica es +

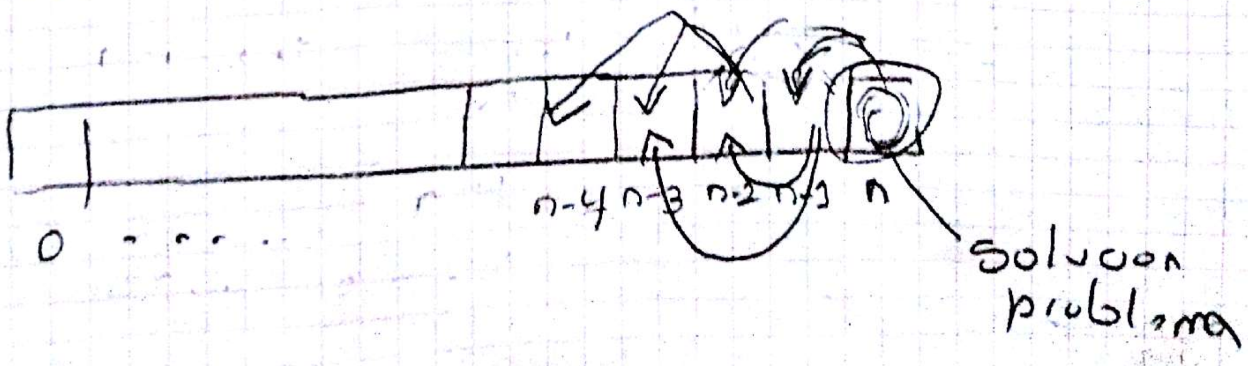
④ Definicon estructura

Arreglo $F[n] = \text{Fibunnaci}[1]$

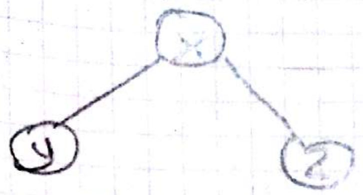


$$F[n] = \begin{cases} 0 & n = 0 \\ 1 & n = 1 \\ F[n] = F[n-1] + F[n-2] & n = 2 \end{cases}$$

⑤ Calculo Forma Bottom-up



⑥ HeapFy   Siempre deja un montón

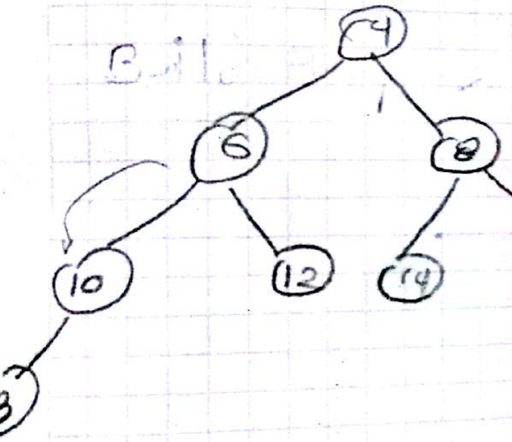


$x \geq y$

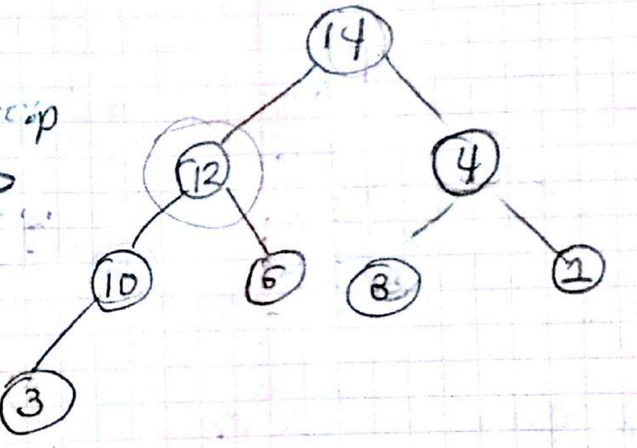
$x \geq z$

No es posible que el primer elemento sea el mayor
Ejemplo

4 6 8 10 12 14 1 3

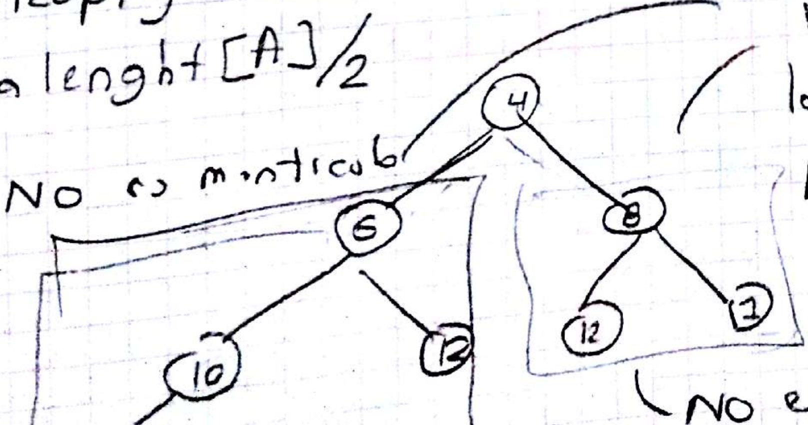


Build Heap $\xrightarrow{\quad}$



Aplicamos HeapFY(i, A)   i = 2 ... n   Queda igual

⑦ HeapFy
1 hasta lenght[A]/2



No se cumple la precondición del HeapFY.

⑧ Algoritmo para ordenar en á n^2 en $O(n)$

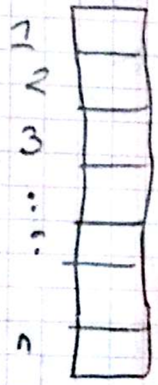
$A = \underbrace{[a_1, a_2, \ldots a_n]}_{n}$

1) Dividimos todo por n^2

$A = [a_1/n^2, a_2/n^2 \ldots a_n/n^2]$ queda entre 0 y 1

2) Aplicamos Bucket-Sort

B[n]



Almacenamos

$B[\lfloor nA[i] \rfloor]$

⑨ SelectionSort

Mejor caso ordenado

Nunca va entra $A[j] < A[min\_ind]$

Complejidad $O(n^2) \rightarrow$ Intercambia elemento consigo mismo

Peor caso. ordenado inversamente

Siempre entra $A[j] < A[min\_ind]$

Complejidad $O(n^2)$

⑩ SelectionSort vs InsertionSort

	Mejor caso	Caso promedio	Peor caso
SelectionSort	$O(n^2)$	$O(n^2)$	$O(n^2)$
InsertionSort	$O(n)$	$O(n^2)$	$O(n^2)$

↳ seleccionó.

(11)

Ingenuo: Dado $B = \{b_1, b_2, \dots b_n\}$
$A = K$

Tomo todos los subconjuntos de $B = O(2^n)$

(12) Si $\longrightarrow$ Si $K - b_i - b_j = 0$ (solucionamos el problema)

$$\begin{tikzcd} & \text{Min} \left(B = (\{b_1, b_2, \dots b_n\}, K) \right) \arrow[dl] \arrow[dr] & \\ \{b_1, b_2, \dots, b_{n-1}, K - b_n\} & & \{b_1, b_2, \dots b_{n-1}\}, K\} \end{tikzcd}$$

Escoger b_n No escoger b_n.

(13) 1) Ordeno de mayor a menor,
2) Elijo hasta completar A

$A = 50$
$B = \{30, 20, 20, 10, 10, 10\}$
$\uparrow \quad \uparrow$

termine $\underline{\{30, 20\}}$

No, una solución voraz **NO** garantiza solución óptima