

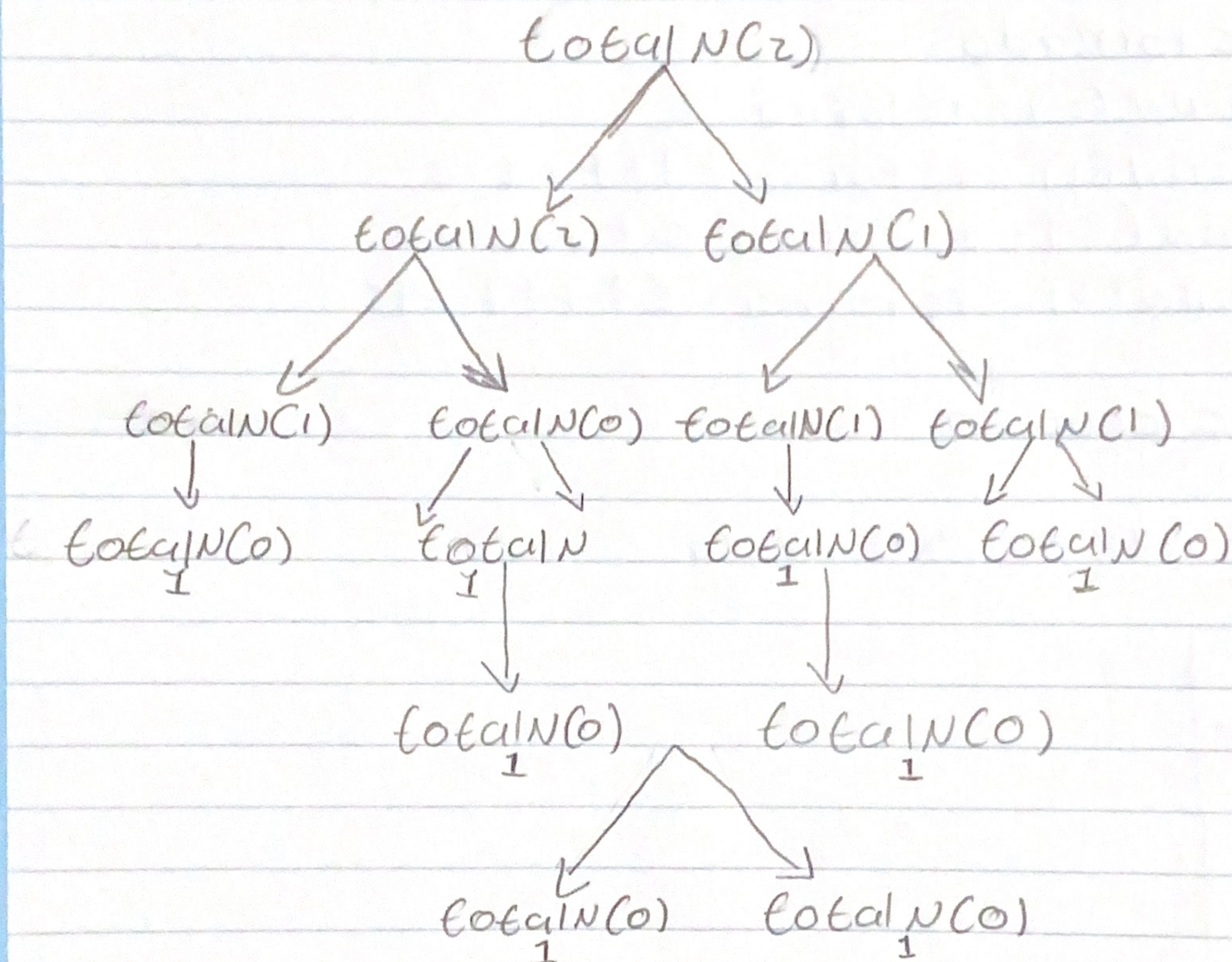
-Codigo

```
public static int totalN(int n) {
```

```
if (n == 0) return 1;
```

```
return 1 + totalN(n-1) + totalN(n-2);
```

- Arbol de recursion



- Ejecucion Paso a Paso

Paso 1 - $\text{CoEa}(N(3))$

Paso 2 - $\text{Cotain}(2)$ - izquierda

Paso 3 - $\text{EofalN}(1)$ - izquierda

Paso 4 - $\text{Eofain}(0) \rightarrow$ caso base

Paso 5 - $\text{Eofain}(0) \rightarrow \text{Caso base}$

Paso 6 - $f(1,1)$ - derecha
 Paso 7 - $f(1,0)$ -> caso base
 Paso 8 - $f(0,0)$ -> caso base
 Paso 9 - $f(2,1)$ - derecha
 Paso 10 - $f(1,1)$ - izquierda
 Paso 11 - $f(1,0)$ -> caso base
 Paso 12 - $f(0,0)$ -> caso base
 Paso 13 - $f(1,1)$ - derecha
 Paso 14 - $f(1,0)$ -> caso base
 Paso 15 - $f(0,0)$ -> caso base

- Fase 2 - Retornado

Paso 1 - $f(0,0)$ - retorna 1
 Paso 2 - $f(1,1)$ - retorna $1+1+1=3$
 Paso 3 - $f(2,1)$ - retorna $1+3+3=7$
 Paso 4 - $f(3,1)$ - retorna $1+7+7=15$

- Pila de ejecución

- Momento 1: máxima profundidad

Pila
$f(0,0)$ ←
Top
$f(2,1)$
$f(2,1)$
$f(3,1)$
Base

- Momento 2 - Después de resolver $f(0,0)$

Pila
$f(0,0)$ ←
Top
$f(3,1)$
Base

pila

vacía

Resultado obtenido = 15

- Ejercicio

Como funciona la recursión en árboles binarios completos, donde calculamos cuantos nodos totales tiene un árbol binario de profundidad.