John Brandon Quiñónez Espinal

1.

```
Void algorithm_A(int n) {
    int count = 0;
    for (int i = 1; i ≤ n; i++) {
        for (int j = i; j ≤ n; j += i) {
            count++;
        }
    }
}
```

a. El bucle externo va de 1 hasta n, el bucle interno empieza en $j = i$ y se incrementa en pasos de i.

Para un i fijo el numero de iteraciones del bucle interno es $\left\lfloor \frac{n}{i} \right\rfloor$, donde $\lfloor k \rfloor$ es el máximo entero de k.

→ el tiempo total

$$T(n) = \sum_{i=1}^{n} \sum_{k=1}^{\left\lfloor \frac{n}{i} \right\rfloor} 1 = \sum_{i=1}^{n} \left\lfloor \frac{n}{i} \right\rfloor$$

b. Como $[[b]] \leq x$

$$\to T(n) = \sum_{i=1}^{n} [[\frac{n}{i}]] \leq \sum_{i=1}^{n} \frac{n}{i}$$

$$\to T(n) = n\left(\sum_{i=1}^{n} \frac{1}{i}\right)$$

c. A partir de la siguiente desigualdad

$$\sum_{i=1}^{n} \frac{1}{i} \leq 1 + \int_{1}^{n} \frac{1}{x}\,dx = 1 + \ln(n)$$

$$\to T(n) \leq n\,(1+\ln(n)) = n + n\ln(n)$$

Como $T(n) \leq n + \frac{1}{\log_2 e}(n)\log_2 n$, por propiedad de logaritmos.

Para $n \geq 2$, Tenemos $n \leq n \log_2 n$

$$\to T(n) \leq n\log_2 n + \frac{1}{\log_2 e}(n)\log_2 n = \left(1 + \frac{1}{\log_2 e}\right) n \log_2 n$$

Tomando $c = \left(1 + \frac{1}{\log_2 e}\right)$ y $g(n) = n \log(n)$

$\to T(n) \leq c\, n \log n$

$\to T(n) \in O(n \log n)$

d. Sirve para calcular una cota superior $\approx$ $\ln(n)$ más pequeña que el trivial n.

2.

a. El bucle externo (i) itera desde 1 hasta n

, el bucle interno (j) itera desde 1 hasta $n-i+1$ para i fijo

, el count++ se ejecuta 1 vez por cada operación interna.

$$\to T(n) = \sum_{i=1}^{n} \sum_{j=1}^{n-i+1} 1$$

b. $\sum_{j=1}^{n-i+1} 1 = (n-i+1)$

$\rightarrow T(n) = \sum_{i=1}^{n} (n-i+1)$

$T(n) = \sum_{i=1}^{n} n+1) - \sum_{i=1}^{n} i$

$T(n) = (n+1)(n) - \frac{n(n+1)}{2}$

$T(n) = \frac{1}{2}n^2 + \frac{1}{2}n$

c. Demostraremos $T(n) \in \Theta(n^2) \rightarrow O(n^2)$

Big-Theta

Primero $O(n^2)$, en efecto

$T(n) = \frac{1}{2}n^2 + \frac{1}{2}n \leq n^2 + \frac{1}{2}n^2 = n^2$

Tomando $C = 1$ y $n_0 = 1$

Ahora Cota inferior $\Omega(n^2)$

$$T(n) = \frac{1}{2}n^2 + \frac{1}{2}n \geq \frac{1}{2}n^2$$

$\rightarrow$ $c = \frac{1}{2}$, $n_0 = 1$

$\rightarrow$ $\frac{1}{2}n^2 \leq T(n) \leq 1 \cdot n^2$, $n \geq 1$

$$T(n) \in \Theta(n^2)$$

$\rightarrow$ La complejidad es cuadrática.

3. a. i se inicializa en n, el while tiene condición de $i > 1$, en cada iteración antes del bucle interno se actualiza a $i = \frac{i}{2}$.

El bucle interno itera j desde 0 hasta $i-1$.

Sea k el índice del iterador del bucle while en la iteración k, se tiene que $i = \frac{n}{2^k}$

$\hookrightarrow$

→ lo que implica que salimos del while en

$\frac{n}{2^k} < 1$, esto es en aproximadamente $\log_2 n$ iteraciones.

$$\to T(n) = \sum_{k=1}^{[[\log_2 n]]} \frac{n}{2^k}$$

b. Se quiere demostrar

$$\sum_{k=0}^{[[\log n]]} \frac{n}{2^k} \leq 2n$$

Basta demostrar

$$\sum_{k=0}^{m} \frac{n}{2^k} \leq 2n \text{, con } m \text{ arbitrario}$$

Ahora

$$S(m) = \sum_{k=0}^{m} \left(\frac{1}{2}\right)^k = 2 - \frac{1}{2^m} \text{ (Hipótesis).}$$

Para caso $m = 0$

$$\to \sum_{k=0}^{0} \left(\frac{1}{2}\right)^k = 1 \quad , \quad 2 - \frac{1}{2^0} = 1$$

$=$ (Cumple caso $m = 0$)

Suponiendo que se cumple caso n

$$\sum_{k=0}^{n} \left(\frac{1}{2}\right)^{k} = 2 - \frac{1}{2^{n}}$$

Demostraremos que se cumple caso $n+1$

$$\rightarrow \sum_{k=0}^{n+1} \left(\frac{1}{2}\right)^{k} = \sum_{k=0}^{n} \left(\frac{1}{2}\right)^{k} + \left(\frac{1}{2}\right)^{n+1}$$

$$= \left(2 - \frac{1}{2^{n}}\right) + \frac{1}{2^{n+1}}$$

$$= 2 - \frac{1}{2^{n+1}}$$

$\Rightarrow$ Cumple para $n+1$, luego se cumple para todo $n \geq 0$, ahora retomando la inecuación original

$$\sum_{k=0}^{n} \frac{n}{2^{k}} = n\left(2 - \frac{1}{2^{n}}\right) = 2n - \frac{n}{2^{n}} < 2n$$

$\rightarrow \displaystyle\sum_{k=0}^{n} \frac{n}{2^{k}} < 2n$, donde n es arbitrario

$\parallel$

$[[\log(n)]]$

c.

Sabemos que es $O(n)$ ~~por~~ a partir de la ~~de~~ desigualdad

$$T(n) = \sum_{k=1}^{[\lceil \log_2^n \rceil]} \frac{n}{2^k} \leq 2n$$

$$c = 2 \quad , n_0 = 1 \implies T(n) \in O(n)$$

Además el primer termino de la serie $(k = 1)$

es $\frac{n}{2} \rightarrow T(n) \geq \frac{n}{2} \rightarrow T(n) \in \Omega(n)$

$$\rightarrow T(n) \in \Theta(n)$$

es complejidad lineal.

4.

En el while para un i fijo, se inicializa $k = 1$, la condición de continuación $k < i$ y se actualiza $k = 2 * k \rightarrow k$ toma $\{2^0, 2^1, 2^2, \ldots, 2^{m-1}\}$

El bucle de la iteración m se detiene cuando el valor $k = 2^m$, ya que no cumple $k < i$.

Por eso, el ultimo k es 2^{m-1}

luego no damos cuenta

$$2^{m-1} < i < 2^m$$

$\rightarrow$ $m-1 < \log_2 i \le m$, por la aplicacion del logaritmo base

$$\rightarrow \lceil \log_2 i \rceil = m$$

$\rightarrow$ el numero de iteraciones para un i fijo es

$$\lceil \log_2 i \rceil .$$

b.

$$T(n) = \sum_{i=1}^{n} \lceil \log_2 i \rceil$$

c. Demostraremos $T(n) \in \Theta(n \log(n))$

Primero $O(n \log(n))$. Como $i \le n \rightarrow$

$$\lceil \log_2 i \rceil \le \lceil \log_2 n \rceil$$

ademas, $\lceil \log_2 i \rceil \le \log_2 i + 1$

Se tiene

$$T(n) = \sum_{i=1}^{n} \lceil \log_2 i \rceil \leq \sum_{i=1}^{n} \log_2 i + 1$$

$$\leq \sum_{i=1}^{n} (\log_2 n + 1)$$

$$\rightarrow T(n) \leq n \log_2 n + n$$

Como $n \leq n \log_2 n$, $n \geq 2$

$$T(n) \leq 2n \log_2 n \Rightarrow T(n) \in O(n \log(n))$$

Ahora cota inferior $\Omega(n \log(n))$

$$c\, g(n) \leq T(n) \quad , \forall n \geq n_0$$

Tomando $c = \frac{1}{4}$, $n_0 = 4$ verifica

$$\frac{1}{4} n \log_2 n \leq \frac{1}{2} n \log_2 n - \frac{1}{2} n$$

Afirmacion se cumple lo anterior.

Ahora de $T(n) = \sum_{i=1}^{n} [[\log_2 i]]$

, Considerando solo la segunda mitad de los terminos de la sumatoria (desde $i = [[\frac{n}{2}]] + 1$, hasta n)

· Para estos terminos

$$i > \frac{n}{2} \rightarrow \log_2 i > \log_2(\frac{n}{2})$$
$$= \log_2(n) - 1$$

$$T(n) = \sum_{i=1}^{n} [[\log_2 i]] \geq \sum_{i=[[n/2]]+1}^{n} [[\log_2 i]]$$

$$\geq \sum_{i=[[\frac{n}{2}]]+1}^{n} \log_2 i \quad , \text{ por } [[x]] \leq x$$

$$\rightarrow \cancel{T(n) \geq \frac{n}{}} \quad T(n) \geq \sum_{i=[[\frac{n}{2}]]+1}^{n} (\log_2 n - 1)$$

$$\rightarrow T(n) \geq \frac{n}{2}(\log_2 n - 1) = \frac{n}{2}\log_2 n - \frac{n}{2}$$

→ Por la afirmacion anterior $c = \frac{1}{4}$, $n_0 = 4$ → $T(n) \in \Omega(n \log n)$

$\rightarrow T(n) \in \Theta(n \log(n))$

5.

a. El numero total de veces que se ejecuta el cuerpo del bucle interno esta determinado por los limites de las sumatorias anidadas.

El bucle externo va de $i=1$ a n

El bucle interno va de $j=1$ a i.

Sea $I(n)$ el total de iteraciones

$$I(n) = \sum_{i=1}^{n} \sum_{j=1}^{i} 1 = \sum_{i=1}^{n} i$$

$$= \frac{n(n+1)}{2} = \frac{1}{2}n^2 + \frac{1}{2}n$$

b. La condicion $(i+j) \% 2 == 0$, ocurre cuando ambos son pares o impares (i, j).

Analizamos la cantidad de valores $j \in \{1, 2, \ldots, i\}$ que cumplen para i fijo. Sea $C(i)$ esta cantidad

→ Si i es par $(i = 2k)$ → j debe ser par

→ Hay $k = \frac{i}{2}$ valores para j

→ Si i es impar $(i = 2k-1)$ → j debe ser impar

→ Hay $k = \frac{(i+1)}{2}$ valores para j

→ $C(i) = \left\lceil \frac{i}{2} \right\rceil$

→ El numero de entradas que cumplen condicion es $E(n) = \sum_{i=1}^{n} \left\lceil \frac{i}{2} \right\rceil$

Para $i = 2k-1$ / $\left\lceil \frac{2k-1}{2} \right\rceil = k$

Para $i = 2k$ / $\left\lceil \frac{2k}{2} \right\rceil = k$

donde $2k-1$ y $2k$ son numeros consecutivos

→ la suma de cada par es $2k$

Si asumimos n par $(n = 2m)$

$$E(n) = \sum_{k=1}^{m} (k+k) = \sum_{k=1}^{m} 2k = m(m+1)$$

Sustituyendo $m = \frac{n}{2}$

$$E(n) = \frac{n}{2}\left(\frac{n}{2}+1\right) = \frac{n^2}{4} + \frac{n}{2}$$

C.- Cota inferior

Para $n \geq 1$, $\frac{1}{2}n > 0$

$$\rightarrow \frac{1}{2}n^2 + \frac{1}{2}n > \frac{1}{2}n^2$$

Tomamos $c_1 = \frac{1}{2} \rightarrow T(n) \in \Omega(n^2)$

Ahora, cota superior

Para $n \geq 1$, $n \leq n^2$

$$\frac{1}{2}n^2 + \frac{1}{2}n \leq \frac{1}{2}n^2 + \frac{1}{2}n^2 = n^2$$

Tomamos $c_2 = 1$

$$\rightarrow T(n) \in O(n^2)$$

$$\rightarrow T(n) \in \Theta(n^2)$$

Complejidad cuadrática

d. Ya demostramos la complejidad Big-O en el paso anterior.

$$T(n) \leq 1 \cdot n^2, \forall n \geq 1$$

$$c = 1 \text{ y } n_0 = 1.$$

$$T(n) \in O(n^2).$$

6. a. El caso base $n \leq 1 \rightarrow T(1) = c_1$, c_1 constante

Caso recursivo cuando se realiza una llamada recursiva con el argumento $\frac{n}{3}$ y suma 1 al resultado. La operación de suma y retorno son constantes c_2.

$$\rightarrow T(n) = T\left(\frac{n}{3}\right) + c.$$

$$, T(1) = c.$$

2.

Iteración 1

$$T(n) = T\left(\frac{n}{3}\right) + c$$

Iteración 2

$$T(n) = \left(T\left(\frac{n}{3^2}\right) + c\right) + c$$

$$= T\left(\frac{n}{3^2}\right) + 2c$$

Iteración 3

$$T(n) = T\left(\frac{n}{3^3}\right) + 3c$$

$\rightarrow$ Iteración K - esimi

$$T(n) = T\left(\frac{n}{3^k}\right) + KC$$

La expansión se detiene cuando se alcanza el caso base

$$\frac{n}{3^k} = 1 \Rightarrow n = 3^k$$

$$k = \log_3 n$$

Ahora reemplazando en la ecuación general

$\rightarrow$ $T(n) = T(1) + (\log_3 n) \cdot c$

$$T(n) = c + c \log_3 n = c(1 + \log_3 n)$$

c. A partir $T(n) = c(1 + \log_3 n)$, sabemos que

$$\log_3 n = \frac{\log_2 n}{\log_2 3}, \text{ como } \frac{1}{\log_2 3} \text{ es constante} \to \log_3 n \in \Theta(\log(n))$$

Tomando $C = c + \dfrac{c}{\log_2 3}$ (para n suficientemente grande dominado por el logaritmo)

$$\to \quad T(n) \leq K \log(n)$$

$$\to \quad T(n) \in O(\log(n))$$

d. La profundidad maxima corresponde al numero de veces que la funcion se llama a si misma antes de llegar al caso base. Ya calculamos el numero "K" de pasos para que el argumento se reduzca de n a 1.

El valor fue $K = \log_3 n$

Si n no es una potencia exacta de 3 luego la profundidad es $K = [[\log_3 n]]$