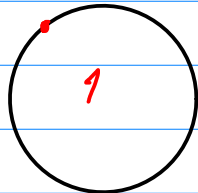


Problema: Con  $n$  puntos en una circunferencia que están unidos por cuerdas, ¿en cuántas piezas está dividido el círculo?

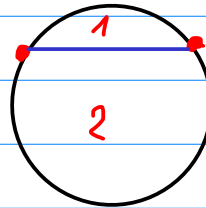
$$n = 1$$

1



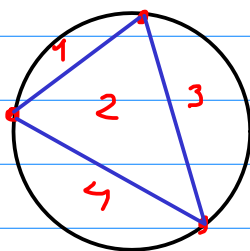
$$n = 2$$

2



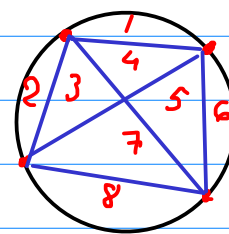
$$n = 3$$

4



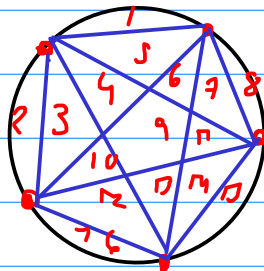
$$n = 4$$

8



$$n = 5$$

16



Aparentemente la respuesta es  $2^{n-1}$ , sin embargo, para  $n=6$  obtenemos 31 piezas, lo cual destruye nuestra hipótesis.

- El incremento de una pieza al añadir una cuerda.  
= una cuerda divide cada pieza en dos.
- La cantidad de piezas divididas por la cuerda  
= cada pieza se divide por un segmento de la cuerda.
- La cantidad de segmentos en la cuerda  
= la cantidad de puntos internos que son intersección con otros segmentos.
- Observamos que  $[1 + \text{cantidad puntos de intersección interna}]$  es la cantidad de piezas que un segmento dividirá en dos.
- El incremento al añadir  $c$  cuerdas  
=  $c + [\text{cantidad total de puntos de intersección interna en las nuevas cuerdas}]$ .

Sea  $f :=$  cantidad de piezas

$c :=$  cantidad de cuerdas

$p :=$  cantidad de puntos de intersección interna

Comenzando con cero puntos, tenemos una pieza, luego

$$f = 1 + c + p.$$

Sea  $n \in \mathbb{N}$  la cantidad de puntos que tenemos de entrada, sabemos que cada cuerda se forma con 2 puntos

$$f(n) = 1 + \binom{n}{2} + p(n)$$

y cada punto de intersección interna se forma con 4 puntos

$$f(n) = 1 + \binom{n}{2} + \binom{n}{4}$$

$$= 1 + \frac{n!}{2!(n-2)!} + \frac{n!}{4!(n-4)!}$$

$$= 1 + \frac{n(n-1)}{2} + \frac{n(n-1)(n-2)(n-3)}{24}$$

$$= 1 + \frac{12(n^2 - n)}{24} + \frac{n^4 - 6n^3 + 11n^2 - 6n}{24}$$

$$= 1 + \frac{n^4 - 6n^3 + 23n^2 - 18n}{24}$$

∴ Con  $n$  puntos en una circunferencia que están unidos por cuerdas, el círculo puede ser dividido en  $1 + \frac{n^4 - 6n^3 + 23n^2 - 18n}{24}$  piezas.

Problema: ¿Cómo calculamos el máximo común divisor de  $A$  y  $B$ ?

• Solución trivial pero ineficiente:

Hallamos los divisores de  $A$  y  $B$

$$\text{div } A = \{ \underset{\text{"i"}}{d_1}, \underset{\text{"A"}}{d_2}, \dots, \underset{\text{"A"}}{d_n} \} \quad \wedge \quad \text{div } B = \{ \underset{\text{"i"}}{e_1}, \underset{\text{"B"}}{e_2}, \dots, \underset{\text{"B"}}{e_m} \}.$$

El cálculo de cada lista sería en  $O(\max(A, B))$ , para cada factor en  $\text{div } A$  realizaría una búsqueda binaria en  $\text{div } B$  para saber si aparece. De

esta manera la complejidad en tiempo sería // Notación Manual  
 $T(A, B) = O(A + B + A \lg B)$  //  $\lg := \log_2$   
 $= O(\max(A, B) \lg(\max(A, B)))$ .

• Solución eficiente: Algoritmo de Euclides

$GCD(A, B)$

if ( $B == 0$ ) return  $A$ ;

else return  $GCD(B, A \bmod B)$

Analicemos la corrección del algoritmo provisto.

Teorema. Para todo  $A \in \mathbb{N} \cup \{0\}$ ,  $B \in \mathbb{N}$  tenemos que  
 $GCD(A, B) = GCD(B, A \bmod B)$ .

Prueba: Sabemos que si  $n \mid m$  y  $m \mid n$ , entonces  $m = n$ , donde  $x \mid y$  significa que  $x$  divide a  $y$ ,  $m, n, x, y \in \mathbb{N} \cup \{0\}$ .

( $\Rightarrow$ ) Comencemos probando que  $GCD(A, B) \mid GCD(B, A \bmod B)$ .

Aplicando el teorema de representación única de la división tenemos

$$A = Bk + r, \quad 0 \leq r < B \quad \wedge \quad r = A \bmod B$$

$$\begin{aligned} r &= A - Bk \quad \text{y como } GCD(A, B) \text{ es un divisor de } A \text{ y } B, \\ \text{tenemos que } &\left. \begin{array}{l} GCD(A, B) \mid A \\ GCD(A, B) \mid Bk \end{array} \right\} GCD(A, B) \mid (A - Bk) \\ &\quad \downarrow \\ &GCD(A, B) \mid (A \bmod B) \end{aligned}$$

$$\begin{aligned} \text{Por lo tanto, } &GCD(A, B) \mid B \quad \text{y} \quad GCD(A, B) \mid (A \bmod B) \\ \Rightarrow &GCD(A, B) \mid GCD(B, A \bmod B). \end{aligned}$$

( $\Leftarrow$ ) Probemos que  $GCD(B, A \bmod B) \mid GCD(A, B)$ .

Análogamente a la prueba anterior, tenemos que

$$GCD(B, A \bmod B) \mid B \quad \wedge \quad GCD(B, A \bmod B) \mid A \bmod B$$

$$A \bmod B = A - Bk \Rightarrow A = Bk + A \bmod B$$

$$\text{Como } GCD(B, A \bmod B) \mid (A \bmod B) \quad \text{y} \quad GCD(B, A \bmod B) \mid Bk$$

$$\text{entonces } GCD(B, A \bmod B) \mid A.$$

$$\text{Como también } GCD(B, A \bmod B) \mid B, \text{ tenemos } GCD(B, A \bmod B) \mid GCD(A, B).$$

◻◻ Concluimos que  $GCD(A, B) = GCD(B, A \bmod B)$  ◻

Teorema.  $\text{GCD}(A, B)$  tiene complejidad en tiempo  $O(\lg(\max(A, B)))$ .

Prueba: Sin pérdida de generalidad, asumamos que  $A \geq B$ . En caso no sería hacemos un swap entre las dos variables. Analicemos dos casos:

i)  $B \leq \frac{A}{2}$ : En este caso, si  $B \neq 0$  tenemos que resolver

$$\text{GCD}(A, B) = \text{GCD}(B, \underbrace{A \bmod B}_{\leq \frac{A}{2}})$$

ii)  $B > \frac{A}{2}$ :  $-B < -\frac{A}{2}$

$$\underbrace{A - B}_{A \bmod B} < \frac{A}{2}$$

$A \bmod B$  pues  $A \geq B$

$$\text{GCD}(A, B) = \text{GCD}(B, \underbrace{A \bmod B}_{< \frac{A}{2}})$$

Notamos que en el caso (i) ambos se reducen al menos a su mitad, mientras que en el caso (ii) solo uno de ellos se reduce al menos a su mitad. Asimismo sabemos que cuando uno de ellos llega a cero, el programa empieza a retornar, por lo cual tenemos que

$$T(A, B) \leq 1 + T\left(A, \frac{B}{2}\right)$$

$$\leq 1 + 1 + T\left(A, \frac{B}{4}\right)$$

:

$$\leq 1 + 1 + \dots + 1 + T\left(A, \frac{B}{2^{\lfloor \lg B \rfloor}}\right)$$

$\leftarrow \lfloor \lg B \rfloor$  veces

: Análogamente con A

$$\leq \underbrace{1 + \dots + 1}_{\lfloor \lg B \rfloor \text{ veces}} + \underbrace{1 + \dots + 1}_{\lfloor \lg A \rfloor \text{ veces}} + T\left(\frac{A}{2^{\lfloor \lg A \rfloor}}, \frac{B}{2^{\lfloor \lg B \rfloor}}\right)$$

C

$$\leq C + \lfloor \lg B \rfloor + \lfloor \lg A \rfloor$$

$$\leq C \lfloor \lg \max(A, B) \rfloor + \lfloor \lg \max(A, B) \rfloor + \lfloor \lg \max(A, B) \rfloor$$

$$\leq (2 + C) \lfloor \lg(\max(A, B)) \rfloor$$

$$\leq (2 + C) (\lg(\max(A, B)) + 1)$$

$$\leq 2(2 + C) \lg(\max(A, B))$$

$$\leq K \lg(\max(A, B))$$

$$\therefore \text{GCD}(A, B) \in O(\lg(\max(A, B))).$$