Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
IIC1253 - Matemáticas Discretas

# Tarea 5

2 de noviembre de 2020

2º semestre 2020 - Profesores G. Diéguez - F. Suárez

---

## Requisitos

- La tarea es individual. Los casos de copia serán sancionados con la reprobación del curso con nota 1,1.
- **Entrega**: Hasta las 23:59:59 del 2 de noviembre a través del buzón habilitado en el sitio del curso (Canvas).
  - Esta tarea debe ser hecha completamente en LaTeX. Tareas hechas a mano o en otro procesador de texto **no serán corregidas**.
  - Debe usar el template LaTeX publicado en la página del curso.
  - Cada problema debe entregarse en un archivo independiente de las demas preguntas.
  - Los archivos que debe entregar son un archivo `PDF` por cada pregunta con su solución con nombre `numalumno-P1.pdf` y `numalumno-P2.pdf` , junto con un `zip` con nombre `numalumno.zip`, conteniendo los archivos `numalumno-P1.tex` y `numalumno-P2.tex` que compilan su tarea. Si su código hace referencia a otros archivos, debe incluirlos también.
- El no cumplimiento de alguna de las reglas se penalizará con un descuento de 0.5 en la nota final (acumulables).
- No se aceptarán tareas atrasadas.
- Si tiene alguna duda, el foro de Canvas es el lugar oficial para realizarla.

# Problemas

## Problema 1 - Correctitud y Complejidad

**Algorithm 1** Encontrar el MCM entre i y j $\in N$, con j $\geq$ i

w = j
**while** mod($w$, i) $\neq 0$ **do**
  $w = w + j$
**end while**
**return** $w$

Para demostrar que el algoritmo es correcto debemos demostrar que (1) este termina, y (2) si termina, es correcto.

1. Para demostrar que el algoritmo se termina, tomemos $x$ como la cantidad de veces que se ha ejecutado el loop. Basta con fijarse en que $x$ nuncá será mayor a $i$, puesto que si $w = j * i$, $mod(w, i) = 0$ y el algoritmo se terminaría. Hemos encontrado una cota superior para $x$, y como este es creciente entonces sabemos que en algún momento el algoritmo termina. Podemos escoger como invariante $w = j * x$ con $x$ la cantidad de veces que se ejecute el loop.

   Demostración:

   **B.I.**: si $x = 1$, $w = j = 1j$.

   **H.I.**: Supongamos que para $x_n$ cualquiera, $w = j * x_n$.

   **T.I**: luego de la iteración $x_n$, tenemos que $w_n = j * x_n$. La siguiente iteración, $w_{n+1} = w_n + j = j * x_n + j = x_{n+1} * j$.

   Y de esta forma nuestro invariante se mantiene independiente del número de iteraciones.

2. Para demostrar que cuando acaba, este es correcto, debemos demostrar que nuestro invariante es el mínimo común multiplo entre $i$ y $j$. Por contradicción, supongamos que este NO es el MCM entre estos dos números. Esto significa que $\exists k$, t.q. $k < w$ y $k$ es el MCM entre $i$ y $j$. En ese caso, la condición del *while* no se cumpliría, pues $mod(k, i) = 0$ puesto que $k$ es divisible por $i$ y el algoritmo se habría acabado retornando $w = k$. De esta forma, el invariante es siempre el menor número divisible por $i$ y $j$.

Para calcular la complejidad del algoritmo, nos basaremos en $i$, es decir el menor número que recibe. Es facil ver que en el mejor caso, $mod(j, i) = 0$, por lo tanto $mod(j * 1, i) = mod(w, i) = 0$ y el algoritmo solamente toma 3 pasos independiende del tamaño de los inputs. Así, en el mejor de los casos nuestro algoritmo es $O(1)$. En el peor de los casos, nos tomará $i$ repeticiones del loop llegar al MCM. De esa forma nuestro algoritmo es $O(i)$ en el peor de los casos. Si ocupamos el numero de digitos de $i$, una buena aproximación sería $d = log_{10} i$ con $d$ número de digitos. si escribimos $i$ en términos del tamaño de digitos,

$i \approx 10^d$, por lo que nuestro algoritmo es $O(10^d)$ en el peor de los casos, es decir se comporta de manera exponencial con respecto al número de digitos de $i$.

## Problema 2 - Notación asintótica

1. Para demostrar que esta afirmación es falsa, podemos demostrar que

$$cn\sqrt{n} < \sqrt{n!} \tag{1}$$

a partir de cierto $n$ y para una constante arbitraria. Si reescribimos la expresión anterior, nos queda

$$cn < \sqrt{(n-1)!} \tag{2}$$

Notemos que la siguiente propiedad se cumple para $n$ a partir de 8:

$$(n-1)! > n^4 \tag{3}$$

y también que $\forall n > c$ con $n \in N_+$

$$n^2 = \sqrt{n^4} > cn \tag{4}$$

por lo tanto,

$$\sqrt{(n-1)!} > cn \tag{5}$$

a partir de $n = 8$ y para cualquier c (con c = 1 nos basta para la demostración). Por lo tanto,

$$\sqrt{n!} \notin O(n\sqrt{n}) \tag{6}$$

2. De manera trivial sabemos que

$$\sqrt{n} < \frac{n}{2} \quad \forall n \in N \land n > 4 \tag{7}$$

por lo tanto,

$$n^{\sqrt{n}} < n^{\frac{n}{2}} \quad \forall n \in N \wedge n > 4 \tag{8}$$

$$n^{\sqrt{n}} < cn^{\frac{n}{2}} \ \ \forall n \in N \wedge n > 4 \wedge c \geq 1 \tag{9}$$

queda demostrado que la propiedad se cumple.