



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
IIC2283 - DISEÑO Y ANÁLISIS DE ALGORITMOS
PROFESOR: NICOLÁS VAN SINT JAN
AYUDANTE: DANTE PINTO

Ayudantía 2

Notación asintótica y ecuaciones de recurrencia

1. Demuestre las siguientes afirmaciones:

- $n! \in \Omega(2^n)$.
- $n^n \notin \mathcal{O}(n!)$.
- $\log(n^n) \in \mathcal{O}(\log(n!))$.

2. El tiempo que demora el algoritmo de **MergeSort** se puede expresar con la siguiente recurrencia:

$$T(n) = \begin{cases} 1 & n = 1 \\ 2 * T(\frac{n}{2}) + f(n) & n > 1 \end{cases}$$

Donde $f(n)$ es el tiempo utilizado por la subrutina **merge**. Asumiendo que $n = 2^k$, resuelva la recurrencia para los siguientes casos:

- El **merge** se implementó como es usual, $f(n) = C \cdot n$, con C una constante tal que $C > 0$
- El **merge** se implementó de forma ineficiente, resultando en $f(n) = C \cdot n^2$, con C una constante tal que $C > 0$.
- El **merge** se implementó de forma innovadora (prácticamente mágica) y funciona en tiempo $f(n) = C$, con C una constante tal que $C > 0$