

# CC4102 - Diseño y Análisis de Algoritmos

## Auxiliar 1

Prof. Jérémy Barbay; Aux. Mauricio Quezada

21 de Marzo de 2012

### 1 Notación Asintótica

1. Defina  $O(\cdot)$ ,  $\Omega(\cdot)$ ,  $\Theta(\cdot)$ ,  $o(\cdot)$  y  $\omega(\cdot)$ . ¿Es cierto que si todo algoritmo resuelve un cierto problema en tiempo  $\Omega(f(n))$  en el peor caso, entonces *ningún* algoritmo que resuelve ese problema corre en tiempo  $o(f(n))$ ?

#### 1.1 Solución

1.  $O(g(n)) = \{f(n) : \exists c > 0, n_0 > 0, \forall n \geq n_0, 0 \leq f(n) \leq cg(n)\}$
2.  $\Omega(g(n)) = \{f(n) : \exists c > 0, n_0 > 0, \forall n \geq n_0, 0 \leq cg(n) \leq f(n)\}$
3.  $o(g(n)) = \{f(n) : \forall c > 0, \exists n_0 > 0, \forall n \geq n_0, 0 \leq f(n) < cg(n)\}$
4.  $\omega(g(n)) = \{f(n) : \forall c > 0, \exists n_0 > 0, \forall n \geq n_0, 0 \leq cg(n) < f(n)\}$

### 2 Cotas inferiores

1. Deduzca la cota inferior del problema de encontrar la posición de un elemento  $x$  en un arreglo ordenado de  $n$  elementos distintos (en el *modelo de comparaciones*) usando:
  - Árboles de decisión, y
  - Estrategia del Adversario.
2. Muestre usando árboles de decisión la cota inferior del problema de ordenamiento de  $n$  elementos distintos de un arreglo desordenado en el *modelo de comparaciones*.
3. Muestre mediante la estrategia del adversario cómo probar una cota inferior de  $\lceil 3n/2 \rceil - 2$  comparaciones para el problema de encontrar el mínimo y el máximo de un arreglo desordenado.

#### 2.1 Solución

1. Usando árboles de decisión, definimos el tiempo de ejecución de un algoritmo como la cantidad de pasos desde la raíz del árbol hasta una hoja. El tiempo en *peor caso* es la profundidad del árbol. El *modelo* corresponde al marco de operaciones que podemos realizar en los algoritmos bajo este modelo. En términos del árbol, corresponde al tipo de preguntas que podemos hacer: ¿Es  $x_i < x_j$ ?, ¿Para cuál índice  $i$  es  $x_i = x$ ?, etc.

En el modelo de comparaciones no nos importan los *valores* de cada elemento del input, sino su orden relativo. En el caso de la búsqueda, tenemos  $n$  elementos y  $n + 1$  posibles respuestas, por lo que el árbol de decisión tendrá  $n + 1$  hojas. El árbol es además binario, pues sólo podemos considerar preguntas del tipo “¿Es  $x_i < x_j$ ?” (o  $>$ ), por lo tanto su altura debe ser al menos  $\lceil \lg n \rceil = \Omega(\log n)$ .

Mediante la estrategia del adversario, la idea es que tenemos un “adversario” el cual puede escoger y modificar el input de un algoritmo de forma de maximizar su trabajo. En el caso de la búsqueda, siempre que hagamos una comparación con el elemento buscado,  $x$ , descartaremos una parte del input. El adversario siempre puede arreglárselas para que la parte mayor sea la que contenga a  $x$ , maximizando el trabajo del algoritmo. En ese caso, nuestra mejor opción es ir comparando de forma de dejar siempre partes iguales, de forma que luego de  $\lg n$  consultas nos quede sólo una alternativa, haciendo en total  $\lg n + 1$  comparaciones. ¿Por qué con el árbol de decisión tenemos sólo  $\lg n$  comparaciones?

2. Un árbol de decisión tendrá  $n!$  hojas, correspondiente a la cantidad de permutaciones posibles del arreglo de  $n$  elementos. Ahora el problema es determinar la permutación correcta (ordenada), lo que corresponde a la altura del árbol de decisión, que es  $\lg n!$ , o  $\Omega(n \log n)$ .
3. (Tabla de comparaciones)

Hay distintas formas de verlo:

- En cantidad de comparaciones, necesitamos  $\lceil n/2 \rceil$  comparaciones para separar en dos conjuntos: potenciales máximos y potenciales mínimos. Luego, se necesitan  $n - 2$  comparaciones para determinar el máximo y el mínimo, dando un total de  $3n/2 - 2$  comparaciones (ver el detalle cuando  $n$  es impar)
- En cantidad de “información”, al comienzo hay  $2n - 2$  posibles candidatos a máximo o mínimo (cada elemento por sí solo puede ser o el máximo o el mínimo) que hay que descartar. Cada comparación entre elementos no comparados anteriormente nos da 2 unidades de información adicionales; luego cada comparación entre estos conjuntos (pot. máxs. y pot. míns.) nos entrega una unidad de información. La primera ronda nos entrega  $n/2 \times 2 = n$  unidades y las  $n - 2$  comparaciones restantes  $n - 2$  unidades, dando en total  $2n - 2$  unidades utilizadas.

### 3 Experimentación y sugerencias para las tareas

1. Estructura del informe: Introducción, Implementación/Algoritmos/Est.Datos, Diseño Experimental, Presentación de Resultados, Análisis, Conclusiones y Anexos.
2. Gráficos (recomendaciones con múltiples gráficos, dimensiones, explicación y escala).
3. Tanto el código como el informe son importantes, pero el informe lo es más.
4. Cosas técnicas:  $\text{\LaTeX}$ (obligatorio), `gnuplot`, `R`, `matlab`, `octave`, etc. (recomendados).