Análisis de Algoritmos

1. Estructuras de Control

Análisis de Algoritmos

(1) Estructuras de control
(2) Barómetro
} Usualmente aplicado a peor caso, son “en detalle”

(3) Análisis del caso promedio
(4) Análisis amortizado
} Usualmente dependiente del “tipo de entrada”

(5) Recurrencias } ¿? Un tipo específico de algoritmos

(6) No veremos análisis asociados a algoritmos específicos *de* estructuras de datos

(7) Diseño + análisis {
- Greedy
- Divide-and-Conquer
- Dynamic programming
- Algoritmos probabilísticos

1. Estructuras de Control

- Secuencias:

P1; P2 $\qquad$ $t1(n)$ y $t2(n)$

$t1;2(n) = t1(n) + t2(n)$

Regla del máximo

$t1;2(n) = t1(n) + t2(n) \in O(\max(t1(n), t2(n)))$

$t1;2(n) = t1(n) + t2(n) \in \Theta(\max(t1(n), t2(n)))$

1. Estructuras de Control

- Condicional (dep. de n asumido):

If (cond)	t1
Then (cuerpo then)	t2
Else (cuerpo else)	t3

Se considera directamente el peor caso:

t1 + max(t2, t3) o directamente

max(t1, t2, t3)

1. Estructuras de Control

- Iteraciones for o ciclos uniformes:

For i $\leftarrow$ 1 to m
    Do P(i)   Puede ser P(i, n), t(i)

  – Si t(i) no depende de i ==> t(i) = t
    - tfor = m t
    - En cualquier caso, identificar #iter
    - tfor = #iter t

1. Estructuras de Control

- Iteraciones for o ciclos uniformes:

For i ← 1 to m
    Do P(i)   Puede ser P(i, n), t(i)

  – Si t(i) depende de i
    - tfor = $\sum_{i=1}^{m} t(i)$
  – Sumas útiles:
    - $\sum_{i=1}^{m} i = \frac{m\,(m+1)}{2}$
    - $\sum_{i=1}^{m} i^2 = \frac{m\,(m+1)\,(2m+1)}{6}$

1. Estructuras de Control

- Iteraciones for o ciclos uniformes:

For $i \leftarrow 1$ to m

Do $P(i)$ Puede ser $P(i, n)$, $t(i)$

- – No confundir “peor caso” con el limite
  - For $j \leftarrow i$ to m o For $j \leftarrow 1$ to i
  - Cantidad de iteraciones: $m-i+1$ o i
  - No hay “peor caso” ($i=1$, $i=m$)

1. Estructuras de Control

- Iteraciones for o ciclos uniformes:

For i $\leftarrow$ 1 to m

Do P(i) Puede ser P(i, n), t(i)

  - Todo lo anterior vale solamente si $1 \leq m$
  - En general: inicio $\leq$ fin

1. Estructuras de Control

- Iteraciones no uniformes
  - while y repeat
  - Cantidad de iteraciones desconocida a priori
  - Dos formas de análisis
    - Funciones de variables que decrece
    - Recurrencias
    - Veremos ambas formas con un ejemplo

1. Estructuras de Control

- Iteraciones no uniformes

```
function bin_search(T[1...n], x)  // x está en T
    i ← 1; j ← n
    While i < j Do          ==> Cantidad determinada por la dif.
        k ← (i + j) % 2
        Case x < T[k]: j ← k-1
             x = T[k]: i, j ← k // Return k
             x > T[k]: i  ← k+1
    Return i
```

(cont.)

1. Estructuras de Control

- Iteraciones no uniformes: función
  - Variables de la iteración
  - El valor de la función decrece a medida que se llevan a cabo más iteraciones
  - El valor de la función debe ser un entero positivo ==> el algoritmo termina
  - ¿Cuándo? Entender la forma en que decrece la función. Ej: bin_search

(cont.)

1. Estructuras de Control

- Iteraciones no uniformes: función
  - bin_search
    - Mejor caso: se encuentra en la 1ra evaluación
    - Peor caso: se encuentra cuando i=j
    - Función d = j - i + 1 (cantidad de elems.)
    - $d \geq 2$ (análisis de "mitad")
    - d = 1 (índice del elem.)

(cont.)

1. Estructuras de Control

- Iteraciones no uniformes: función
  - bin_search: $d = j - i + 1$

¿Decrece? Veamos los valores de i, j y d al principio y al final de una iteración cualquiera, i', j', y d‘:

1) $x < T[k]$:

$j' = (i + j) \div 2 - 1; \quad i' = i;$

$d' = (i + j) \div 2 - 1 - i + 1 \leq (i + j) / 2 - i$

$= - i/2 + j/2 < - i/2 + j/2 + 1/2 = (j - i + 1) / 2 = d/2 < d$

(cont.)

1. Estructuras de Control

- Iteraciones no uniformes: función
  - bin_search: $d = j - i + 1$

¿Decrece? Veamos los valores de i, j y d al principio y al final de una iteración cualquiera, i', j', y d':

2) $x > T[k]$:

$$j' = j; \quad i' = (i + j) \div 2 + 1;$$

$$d' = j - (i + j) \div 2 - 1 + 1 \leq j - (i + j) / 2 =$$

$$= j/2 - i/2 < j/2 - i/2 + 1/2 = (j - i + 1) / 2 = d/2 < d$$

(cont.)

1. Estructuras de Control

- Iteraciones no uniformes: función
  - bin_search: $d = j - i + 1$

  ¿Decrece? Veamos los valores de i, j y d al principio y al final de una iteración cualquiera, i', j', y d‘:

  3) $x = T[k]$:

  $d' = 1 \leq d/2$

(cont.)

==> $d = j-i+1$decrece en cada iteración

1. Estructuras de Control

- Iteraciones no uniformes: función
  - bin_search: $d = j - i + 1$

  ¿Cuántas iteraciones? d_k: valor de $j_k - i_k + 1$ al final de la iteración k

  $d_0 = n$

  $d_1 = n/2$

  ...

  $d_i = n/2^i$

  Peor caso, “todas” las iteraciones hasta que $d_k = 1$

  $d_k = n/2^k = 1$; ¿k? $n = 2^k \implies \log_2 n = \log_2 2^k \implies$

  $k = \log_2 2^n \implies \lceil k = \log_2 n \rceil$

1. Estructuras de Control

- Iteraciones no uniformes
  - Dos formas de análisis
    - Funciones de variables que decrece
      - Explicación y ejemplo bin_search
    - Recurrencias

1. Estructuras de Control

- Iteraciones no uniformes: recurrencias
  - t(n): t para resolver el problema con n elems.
  - $t(n) \begin{cases} 1 & n = 1 \\ t(n/2) + c & n > 1 \end{cases}$
  - Veremos las recurrencias más adelante
- En general, las iteraciones no uniformes son complicadas de analizar (función y/o recurrencia)

1. Estructuras de Control

- Resumen:
  - Paso a paso, muy detallado
  - Iteraciones no uniformes complicadas
  - t(n) relativamente detallado y “combinado”
  - Puede implicar mucho tiempo por el detalle de cada estructura de control