El Teorema Maestro

Muchas de las ecuaciones de recurrencia que vamos a usar en este curso tienen la siguiente forma:

$$T(n) = \begin{cases} c & n = 0 \\ a \cdot T(\lfloor \frac{n}{b} \rfloor) + f(n) & n \geq 1 \end{cases}$$

donde a, b y c son constantes, y $f(n)$ es una función arbitraria.

El Teorema Maestro nos dice cuál es el orden de $T(n)$ dependiendo de ciertas condiciones sobre a, b y $f(n)$

El Teorema Maestro

El Teorema Maestro también se puede utilizar cuando $\lfloor \frac{n}{b} \rfloor$ es reemplazado por $\lceil \frac{n}{b} \rceil$

Antes de dar el enunciado del Teorema Maestro necesitamos definir una condición de regularidad sobre la función $f(n)$

Una condición de regularidad sobre funciones

Dado: función $f : \mathbb{N} \to \mathbb{R}_0^+$ y constantes $a, b \in \mathbb{R}$ tales que $a \geq 1$ y $b > 1$

Definición

f es (a, b)-regular si existen constantes $c \in \mathbb{R}^+$ y $n_0 \in \mathbb{N}$ tales que $c < 1$ y

$$(\forall n \geq n_0)(a \cdot f(\lfloor \frac{n}{b} \rfloor) \leq c \cdot f(n))$$

Ejercicio

1. Demuestre que las funciones n, n^2 y 2^n son (a, b)-regulares si $a < b$.
2. Demuestre que la función $\log_2(n)$ no es (1,2)-regular.

Una solución al segundo problema

Por contradicción, supongamos que $\log_2(n)$ es (1,2)-regular.

Entonces existen constantes $c \in \mathbb{R}^+$ y $n_0 \in \mathbb{N}$ tales que $c < 1$ y

$$(\forall n \geq n_0)(\log_2(\lfloor \frac{n}{2} \rfloor) \leq c \cdot \log_2(n))$$

De esto concluimos que:

$$(\forall k \geq n_0)(\log_2(\lfloor \frac{2 \cdot k}{2} \rfloor) \leq c \cdot \log_2(2 \cdot k))$$

Una solución al segundo problema

Vale decir:

$$(\forall k \geq n_0)(\log_2(k) \leq c \cdot (\log_2(k) + 1))$$

Dado que $0 < c < 1$, concluimos que:

$$(\forall k \geq n_0)(\log_2(k) \leq \frac{c}{1-c})$$

Lo cual nos lleva a una contradicción. $\square$

El enunciado del Teorema Maestro

Teorema

Sea $f : \mathbb{N} \to \mathbb{R}_0^+$, $a, b, c \in \mathbb{R}_0^+$ *tales que* $a \geq 1$ *y* $b > 1$, *y* $T(n)$ *una función definida por la siguiente ecuación de recurrencia:*

$$T(n) \quad = \quad \begin{cases} c & n = 0 \\ a \cdot T(\lfloor \frac{n}{b} \rfloor) + f(n) & n \geq 1 \end{cases}$$

Se tiene que:

1. *Si* $f(n) \in O(n^{\log_b(a) - \varepsilon})$ *para* $\varepsilon > 0$, *entonces* $T(n) \in \Theta(n^{\log_b(a)})$
2. *Si* $f(n) \in \Theta(n^{\log_b(a)})$, *entonces* $T(n) \in \Theta(n^{\log_b(a)} \cdot \log_2(n))$
3. *Si* $f(n) \in \Omega(n^{\log_b(a) + \varepsilon})$ *para* $\varepsilon > 0$ *y* f *es* (a, b)*-regular, entonces* $T(n) \in \Theta(f(n))$

Usando el Teorema Maestro

Considere la siguiente ecuación de recurrencia:

$$T(n) \quad = \quad \begin{cases} 1 & n = 0 \\ 3 \cdot T(\lfloor \frac{n}{2} \rfloor) + c \cdot n & n \geq 1 \end{cases}$$

Dado que $\log_2(3) > 1.5$, tenemos que $\log_2(3) - 0.5 > 1$

Deducimos que $c \cdot n \in O(n^{\log_2(3) - 0.5})$, por lo que usando el Teorema Maestro concluimos que $T(n) \in \Theta(n^{\log_2(3)})$

El Teorema Maestro y la función $\lceil x \rceil$

Suponga que cambiamos $\lfloor \frac{n}{b} \rfloor$ por $\lceil \frac{n}{b} \rceil$ en la definición de (a, b)-regularidad.

Entonces el Teorema Maestro sigue siendo válido pero con $T(\lfloor \frac{n}{b} \rfloor) + f(n)$ reemplazado por $T(\lceil \frac{n}{b} \rceil) + f(n)$

Analizando la complejidad de un algoritmo

Sea $\mathcal{A} : \Sigma^* \to \Sigma^*$ un algoritmo

- Recuerde que $t_{\mathcal{A}}(n)$ es el mayor número de pasos realizados por $\mathcal{A}$ sobre las entradas $w \in \Sigma^*$ de largo n

Definición (Complejidad en el peor caso)

Decimos que $\mathcal{A}$ en el peor caso es $O(f(n))$ si $t_{\mathcal{A}}(n) \in O(f(n))$

Analizando la complejidad de un algoritmo

Hasta ahora sólo hemos analizado la complejidad de un algoritmo considerando el peor caso.

También tiene sentido estudiar la complejidad del algoritmo $\mathcal{A}$ en el caso promedio, el cual está dado por la siguiente expresión:

$$\frac{1}{k^n} \cdot \left(\sum_{w \in \Sigma^n} \text{tiempo}_{\mathcal{A}}(w) \right)$$

donde $k = |\Sigma|$ y $\Sigma^n = \{w \in \Sigma^* \mid |w| = n\}$

El caso promedio de un algoritmo

La definición anterior asume que todas las entradas son igualmente probables.

- Esto podría no reflejar la distribución de las entradas en la práctica

Para solucionar este problema podemos usar otras distribuciones de probabilidad.

El caso promedio de un algoritmo

Suponemos que para cada $n \in \mathbb{N}$ hay una distribución de probabilidades:

$\Pr_n(w)$ es la probabilidad de que $w \in \Sigma^n$ aparezca como entrada de $\mathcal{A}$

Nótese que $\displaystyle\sum_{w \in \Sigma^n} \Pr_n(w) = 1$

Ejemplo

Para la definición de caso promedio en las transparencias anteriores tenemos que $\Pr_n(w) = \frac{1}{k^n}$ con $k = |\Sigma|$

El caso promedio de un algoritmo

Para definir el caso promedio, para cada $n \in \mathbb{N}$ usamos una variable aleatoria X_n

- Para cada $w \in \Sigma^n$ se tiene que $X_n(w) = \mathit{tiempo}_{\mathcal{A}}(w)$

Para las entradas de largo n, el número de pasos de $\mathcal{A}$ en el caso promedio es el valor esperado de la variable aleatoria X_n:

$$E(X_n) \quad = \quad \sum_{w \in \Sigma^n} X_n(w) \cdot \Pr_n(w)$$

Definición (Complejidad en el caso promedio)

Decimos que $\mathcal{A}$ en el caso promedio es $O(f(n))$ si $E(X_n) \in O(f(n))$

Sobre las definiciones de peor caso y caso promedio

Notación

Las definiciones de peor caso y caso promedio pueden ser modificadas para considerar las notaciones Θ y Ω

- Simplemente reemplazando $O(f(n))$ por $\Theta(f(n))$ u $\Omega(f(n))$, respectivamente

Por ejemplo, decimos que $\mathcal{A}$ en el caso promedio es $\Theta(f(n))$ si $E(X_n) \in \Theta(f(n))$

Un ejemplo: el algoritmo de ordenación Quicksort

Quicksort es un algoritmo de ordenación muy utilizado en la práctica.

La función clave para la definición de Quicksort:

Partición(L, m, n)
    $pivote := L[m]$
    $i := m$
    for $j := m + 1$ **to** n **do**
        if $L[j] \leq pivote$ **then**
            $i := i + 1$
            intercambiar $L[i]$ con $L[j]$
    intercambiar $L[m]$ con $L[i]$
    return i

Nótese que en la definición de **Partición** la lista L es pasado por referencia

- Las listas (o arreglos) siempre son pasados por referencia en este curso

Un ejemplo: el algoritmo de ordenación Quicksort

La definición de Quicksort:

Quicksort(L, m, n)
  if $m < n$ **then**
    $\ell :=$ **Partición**(L, m, n)
    Quicksort(L, m, $\ell - 1$)
    Quicksort(L, $\ell + 1$, n)

¿Cuál es la complejidad de Quicksort?

Vamos a contar el número de comparaciones al medir la complejidad de **Quicksort**.

- ¿Es ésta una buen medida de complejidad?

Sí **Partición** siempre divide a una lista en dos listas del mismo tamaño, entonces la complejidad de **Quicksort** estaría dada por la siguiente ecuación de recurrencia:

$$T(n) \quad = \quad \begin{cases} 1 & n = 0 \\ 2 \cdot T(\lfloor \frac{n}{2} \rfloor) + c \cdot n & n \geq 1 \end{cases}$$

¿Cuál es la complejidad de Quicksort?

Dado que $c \cdot n \in \Theta(n^{\log_2(2)})$, concluimos usando el Teorema Maestro que $T(n) \in \Theta(n \cdot \log_2(n))$

¿Pero qué nos asegura que **Partición** divide a una lista en dos listas del mismo tamaño?

Ejercicio

1. Dada una lista con n elementos, ¿cuál es el peor caso para **Quicksort**?
2. Demuestre que **Quicksort** en el peor caso es $\Theta(n^2)$

¿Por qué usamos Quicksort entonces?

Vamos a demostrar que **Quicksort** en el caso promedio es $\Theta(n \cdot \log_2(n))$

- Suponiendo una distribución uniforme en las entradas

Vamos a considerar listas sin elementos repetidos.

- Usted va a tener que pensar cómo obtener la misma complejidad para listas con elementos repetidos

La complejidad de Quicksort en el caso promedio

Sea $n \in \mathbb{N}$ tal que $n \geq 2$, y sea $\mathcal{E}_n$ el conjunto de listas L con n elementos distintos sacados desde el conjunto $\{1, \ldots, n\}$

- Tenemos que $|\mathcal{E}_n| = n!$

Para cada $L \in \mathcal{E}_n$ tenemos lo siguiente:

- $\text{Pr}_n(L) = \frac{1}{n!}$
- $X_n(L)$ es el número de comparaciones realizadas por la llamada **Quicksort**(L, 1, n)

La complejidad de Quicksort en el caso promedio

La complejidad de **Quicksort** en el caso promedio está dada por $E(X_n)$

¿Por qué nos restringimos a las listas con elementos sacados desde el conjunto $\{1, \ldots, n\}$?

- ¿En qué sentido estamos considerando todas las listas posibles con n elementos (sin repeticiones)?
- ¿Cómo refleja esto el hecho de que estamos considerando las entradas de un cierto largo?

Calculando el valor esperado de X_n

Sea $L \in \mathcal{E}_n$

Para cada $i, j \in \{1, \ldots, n\}$ con $i \leq j$ defina la siguiente variable aleatoria:

$Y_{i,j}(L)$: número de veces que i es comparado con j en la llamada **Quicksort**(L, 1, n)

Entonces tenemos lo siguiente:

$$X_n(L) = \sum_{i=1}^{n} \sum_{j=i}^{n} Y_{i,j}(L)$$

Calculando el valor esperado de X_n

Dado que el valor esperado de una variable aleatoria es una función lineal, concluimos que:

$$E(X_n) = \sum_{i=1}^{n} \sum_{j=i}^{n} E(Y_{i,j})$$

Para calcular $E(X_n)$ basta entonces calcular $E(Y_{i,j})$ para cada $i, j \in \{1, \ldots, n\}$ con $i \leq j$

Calculando el valor esperado de $Y_{i,j}$

Por la definición de **Partición** no es posible comparar un elemento consigo mismo, por lo que $Y_{i,i}(L) = 0$ para cada $i \in \{1, \ldots, n\}$ y $L \in \mathcal{E}_n$

- Tenemos entonces que $E(Y_{i,i}) = 0$

Consideremos ahora el caso $i = 1$ y $j = n$

Los elementos 1 y n sólo pueden ser comparados en la llamada **Partición**(L, 1, n)

- Estos elementos son comparados si $L[1] = 1$ o $L[1] = n$
- Se realiza a lo más una comparación entre ellos

Tenemos entonces que $Y_{1,n}$ es igual a 0 ó 1

Calculando el valor esperado de $Y_{i,j}$

Además, $\Pr(Y_{1,n} = 1)$ es igual a la probabilidad de que $L[1] = 1$ o $L[1] = n$ dado que L es escogido al azar y con distribución uniforme desde el conjunto $\mathcal{E}_n$

Tenemos entonces que:

$$\Pr(Y_{1,n} = 1) \;=\; \frac{2 \cdot (n-1)!}{n!} \;=\; \frac{2}{n}$$

Nótese que esta probabilidad puede cambiar si consideramos otra distribución de probabilidades sobre las listas en $\mathcal{E}_n$

Concluimos que:

$$E(Y_{1,n}) \;=\; 0 \cdot \Pr(Y_{1,n} = 0) + 1 \cdot \Pr(Y_{1,n} = 1) \;=\; \frac{2}{n}$$

Calculando el valor esperado de $Y_{i,j}$

Consideramos ahora el caso general $1 \leq i < j \leq n$

Mientras las llamadas a **Partición** escojan un valor para *pivote* tal que $pivote < i$ o $pivote > j$, los elementos i y j no son comparados y están en una parte de la lista que va a ser ordenada por **Quicksort**.

- i y j pueden ser comparados en las siguientes llamadas a **Partición**

Calculando el valor esperado de $Y_{i,j}$

Si **Partición** escoge un valor para *pivote* tal que $i < pivote < j$, entonces i no es comparado con j en la ejecución completa de **Quicksort**.

La única forma en que **Quicksort** puede comparar i con j es que el primer elemento que escoja **Partición** desde el conjunto $\{i, i+1, \ldots, j\}$ sea i ó j

- Se realiza a lo más una comparación entre i y j en la ejecución completa de **Quicksort**

Calculando el valor esperado de $Y_{i,j}$

Tenemos entonces que $Y_{i,j}$ es igual a 0 ó 1, y además que:

$$\Pr(Y_{i,j} = 1) \;=\; \frac{2}{j - i + 1}$$

Concluimos que:

$$E(Y_{i,j}) \;=\; 0 \cdot \Pr(Y_{i,j} = 0) + 1 \cdot \Pr(Y_{i,j} = 1) \;=\; \frac{2}{j - i + 1}$$

El cálculo final

Concluimos que:

$$\begin{aligned}
E(X_n) &= \sum_{i=1}^{n} \sum_{j=i}^{n} E(Y_{i,j}) \\
&= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E(Y_{i,j}) \\
&= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1} \\
&= \sum_{i=1}^{n-1} \sum_{k=2}^{n-i+1} \frac{2}{k} \\
&= \sum_{k=2}^{n} (n+1-k) \cdot \frac{2}{k} \\
&= 2 \cdot (n+1) \cdot \left(\sum_{k=2}^{n} \frac{1}{k} \right) - 2 \cdot (n-1) \\
&= 2 \cdot (n+1) \cdot \left(\sum_{k=1}^{n} \frac{1}{k} \right) - 4 \cdot n
\end{aligned}$$

La sumatoria armónica

Para terminar el calculo de $E(X_n)$ tenemos que acotar la sumatoria armónica $\sum_{k=1}^{n} \frac{1}{k}$

Tenemos que:

$$\sum_{k=2}^{n} \frac{1}{k} \ \leq \ \int_{1}^{n} \frac{1}{x}\, dx \ \leq \ \sum_{k=1}^{n} \frac{1}{k}$$

Dado que $\int_1^n \frac{1}{x}\, dx = \ln(n) - \ln(1) = \ln(n)$, concluimos que:

$$\ln(n) \ \leq \ \sum_{k=1}^{n} \frac{1}{k} \ \leq \ \ln(n) + 1$$

La sumatoria armónica

Por lo tanto:

$$2 \cdot (n+1) \cdot \ln(n) - 4 \cdot n \;\leq\; E(X_n) \;\leq\; 2 \cdot (n+1) \cdot (\ln(n)+1) - 4 \cdot n$$

De lo cual concluimos que $E(X_n) \in \Theta(n \cdot \log_2(n))$

- Vale decir, **Quicksort** en el caso promedio es $\Theta(n \cdot \log_2(n))$