Análisis de caso promedio

Parte II

Segundo semestre 2022

IIC2283

Prof. Nicolás Van Sint Jan

Recordatorio: Caso promedio

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

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

Para definir el caso promedio, para cada $n \in \mathbb{N}$ usamos una **variable aleatoria** X_n tal que para cada $w \in \Sigma^n$ se tiene que

$$X_n(w) = tiempo_{\mathcal{A}}(w)$$

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:

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

Definición: Complejidad en el caso promedio

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

Recordatorio: 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**.

Recordatorio: 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)

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

Outline

Complejidad de Quicksort

¿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**.

- Queda como ejercicio el pensar en 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:

- $\mathrm{Pr}_n(L) = \frac{1}{n!}$
- $X_n(L)$ 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:

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

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 $i < \mathit{pivote} < j$, 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:

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

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 $\Theta(n \cdot \log_2(n))$

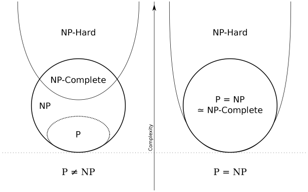
FIN . . .

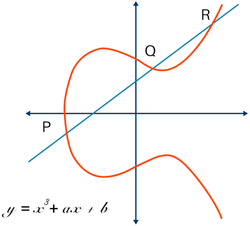
FIN . . .
¿dónde puedo saber más sobre teoría de la computación?

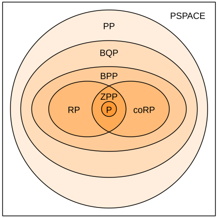
Algunos cursos de postgrado (IIC3xxx):

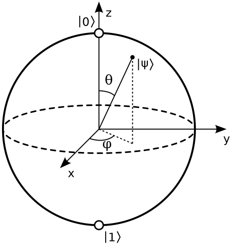
1. Complejidad computacional.
2. Criptografía y seguridad computacional.
3. Tópicos avanzados en ciencias de la computación.
4. Teoría de modelos finitos.

FIN









Si están interesados, solo preguntar!