Diseño y Análisis de Algoritmos **Nombre:** *Sergio Montoya*
Tarea 3
24 de agosto de 2024

Punto 2

Código

```
    public static int maxMatrix(int[][] matrix,
          int from_x,
          int to_x,
          int from_y,
          int to_y) {
    // Base                                                    // Cost
| Time
      if (to_x - from_x < 1 && to_y - from_y < 1)
      return matrix[to_x][to_y];                              // C1 | 1
      int mid_x = from_x + (to_x - from_x)/2;                  // C2 | 1
      mid_x = Math.min(mid_x, to_x - 1);                      // C3 | 1
      int mid_y = from_y + (to_y - from_y)/2;                  // C4 | 1
      mid_y = Math.min(mid_y, to_y - 1);                      // C5 | 1
      // Divide
      int val1 =
      maxMatrix(matrix, from_x, mid_x, from_y, mid_y);        // F(n/4)| 1
      int val2 =
      maxMatrix(matrix, mid_x + 1, to_x, from_y, mid_y);      // F(n/4)| 1
      int val3 =
      maxMatrix(matrix, from_x, mid_x, mid_y + 1, to_y);      // F(n/4)| 1
      int val4 =
      maxMatrix(matrix, mid_x + 1, to_x, mid_y + 1, to_y);    // F(n/4)| 1
      // Conquer
      return
      Math.max(Math.max(val1, val2), Math.max(val3, val4));   // C6 | 1
    }
```

Análisis

Con esto entonces podemos saber que la ecuación con la que vamos a trabajar es:

$$F(n) = \begin{cases} C_1 & n \leq 1 \\ 4F\left(\frac{n}{4}\right) + K & n > 1 \end{cases}.$$

Con esto entonces podemos hacer un intercambio con $n = 4^m$ por lo cual nos queda:

$$F\left(4^m\right) = 4F\left(4^{m-1}\right) + K.$$

Ademas podemos sustituir $F\left(4^m\right) = G\left(m\right)$ con lo cual seria:

$$G\left(m\right) = 4G\left(m-1\right) + k.$$

Y a partir de aquí podemos trabajar con relativa normalidad.

$$\begin{aligned}
G\left(m\right) - 4G\left(m-1\right) &= k \\
\lambda - 4 &= 0 \\
\lambda &= 4 \\
H\left(m\right) &= r_1 \cdot 4^m \\
\text{Hipótesis Sol. Particular: } P(m) &= r_2 \\
G\left(m\right) - 4G\left(m-1\right) &= k \\
r_2 - 4r_2 &= k \\
-3r_2 &= k \\
r_2 &= -\frac{k}{3} \\
&.
\end{aligned}$$

Recuerde que en teoría k esta definido como la suma de las constantes. Por lo tanto, este es un resultado valido. No necesitamos calcular las constantes pues no conocemos los valores precisos del lenguaje y no vale la pena sacarlos con Benchmarks entonces simplemente calcularemos su forma O.

$$\begin{aligned}
G\left(m\right) &= r_1 \cdot 4^m + r_2 \\
m &= \log_4\left(n\right) \\
F\left(n\right) &= r_1 \cdot n + r_2 \\
F\left(n\right) &= O\left(n\right).
\end{aligned}$$

Nota:

En este punto n lo definimos como el total de elementos que hay en la matriz pues no sabemos como trabajar con dos variables en ecuaciones de recurrencia. Sin embargo, esta es una aproximación útil pues la función se comporta cumpliendo esta relación.

Punto 4

Código

```
public static void sort(int[] a, int[] aux, int lo, int hi) {
  // Base                                                  Cost   | Times
  if (hi - lo < 2) return;                                \\ C1     | 1
  int tird = lo + (hi - lo)/3;                            \\ C2     | 1
  int stird = lo + 2 * ((hi - lo)/3) + 1;                 \\ C3     | 1
  // Divide
  sort(aux, a, tird, stird);                              \\ T(n/3)| 1
  sort(aux, a, lo  , tird);                               \\ T(n/3)| 1
  sort(aux, a, stird, hi);                                \\ T(n/3)| 1
  // Conquer
  merge(aux, a, lo, tird, stird, hi);                     \\ O(n)   | 1
}
```

Análisis

Como podemos ver en este caso por la enumeración que hicimos antes la ecuación total queda

$$T(n) = \begin{cases} C_1 & n \leq 1 \\ 3T\left(\frac{n}{3}\right) + a * n + k & n > 1 \end{cases}.$$

Con lo cual tenemos que sustituir con $n = 3^m$ lo que nos deja en

$$T(3^m) = 3T\left(3^{m-1}\right) + a * 3^m + k.$$

Ahora reemplazamos con $G(m) = T(3^m)$

$$G(m) = 3G(m-1) + a * 3^m + k.$$

Con esto ahora podemos aplicar los pasos mas típicos para calcular el resultado de una ecuación recurrente:

$$\begin{aligned} G(m) - 3G(m-1) &= a * 3^m + k \\ \lambda - 3 &= 0 \\ \lambda &= 3 \\ H(m) &= r_1 * 3^m \\ \text{Hipótesis Sol. Particular: } P(m) &= m * r_2 * a * 3^m + k \\ r_2 * m * a * 3^m + k - 3 \cdot m - 1 * a * 3^{m-1} - k &= a * 3^m + k \\ m * a * 3^m + k - (m-1) * a * 3^m - k &= a * 3^m + k \\ a * 3^m (mr_2 - mr_2 + 1) &= a * 3^m + k \\ a * 3^m (1-1) &= k \\ k &= 0 \\ G(m) &= r_1 * 3^m + m * r_2 * a * 3^m. \end{aligned}$$

No nos hace falta llegar hasta los ultimos resultados pues ya tenemos lo suficiente para calcular $O(n)$ por lo tanto, solo nos falta cambiar por $m = \log_3(n)$ y sacar su notación O

$$\begin{aligned} G(m) &= r_1 * 3^{\log_3(n)} + \log_3(n) * r_2 * a * 3^{\log_3(n)} \\ T(n) &= r_1 * n + \log_3(n) * n * r_2 * a \\ T(n) &= O(n \log_3(n)). \end{aligned}$$