Máximo recursivo

- Sea $A_1, \ldots, A_n$ un arreglo con n enteros y considere la función máximo(n) que regresa A_1 si $n \leq 1$ y máx(máximo($n-1$), A_n) en caso contrario.
- Demostremos por inducción que máximo(n) regresa máx($A_1, \ldots, A_n$).
- Para $n = 1$ la llamada máximo(n) regresa A_1.
- Suponga que $n \geq 1$ y que máximo(n) regresa máx($A_1, \ldots, A_n$).
- Entonces, la llamada máximo($n+1$) regresa

$$\text{máx}(\text{máximo}(n), A_{n+1})$$

es decir

$$\text{máx}(\text{máx}(A_1, \ldots, A_n), A_{n+1})$$

o bien

$$\text{máx}(A_1, \ldots, A_n, A_{n+1}).$$

Máximo iterativo

Función máximo(n)

1. $m \leftarrow A_1$, $i \leftarrow 2$.
2. Mientras $i \leq n$ haz:
   1. Si $A_i > m$ entonces $m \leftarrow A_i$.
   2. $i \leftarrow i + 1$.
3. Regresa m.

- Demostraremos que máximo(n) regresa máx($A_1, \ldots, A_n$).

Valores de las variables

- La variable i cumple que $i_0 = 2$ e $i_{j+1} = i_j + 1$.
- La variable m cumple que $m_0 = A_1$ y $m_{j+1} = \text{máx}(m_j, A_{i_j})$.

Ejemplo con $n = 4$ y $A = (3, 1, 4, 1)$

j	i_j	m_j
0	2	3
1	3	3
2	4	4
3	5	4

Invariante y demostración

- Invariante: $i_j = j + 2$ y $m_j = \text{máx}(A_1, \ldots, A_{j+1})$.
- Lo demostraremos por inducción en j.
- Para $j = 0$ tenemos $i_0 = 2$ y $m_0 = \text{máx}(A_1)$.
- Suponga que $j \geq 0$, $i_j = j + 2$ y $m_j = \text{máx}(A_1, \ldots, A_{j+1})$. Entonces $i_{j+1} = i_j + 1 = (j+2) + 1 = (j+1) + 2$ y

$$\begin{aligned} m_{j+1} &= \text{máx}(m_j, A_{i_j}) \\ &= \text{máx}(m_j, A_{j+2}) \\ &= \text{máx}(\text{máx}(A_1, \ldots, A_{j+1}), A_{j+2}) \\ &= \text{máx}(A_1, \ldots, A_{j+1}, A_{j+2}). \end{aligned}$$

- Que es lo que queríamos demostrar.

Correctitud

- Demostraremos que el algoritmo termina con m conteniendo el valor máximo en $A_1, \ldots, A_n$.
- El algoritmo termina cuando $i_j = n + 1$.
- Por el invariante $i_j = j + 2$ esto ocurre cuando $j = n - 1$.
- El valor final de m es $m_{n-1} = \text{máx}(A_1, \ldots, A_n)$.

Máximo y mínimo por separado

- Considere el arreglo de enteros $a = (a_1, a_2, \ldots, a_n)$.
- Se puede encontrar el máximo de a con $n - 1$ comparaciones.
- Se puede encontrar el mínimo de a con $n - 1$ comparaciones.
- Por lo tanto se puede encontrar el máximo y el mínimo del arreglo a con $2n - 2 \in O(n)$ comparaciones.
- ¿Se podrán encontrar con menos de $O(n)$ comparaciones?
- ¿Se podrán encontrar con menos de $2n - 2$ comparaciones?

Máximo y mínimo simultaneamente

- Divida el arreglo a la mitad.
- Encuentre recursivamente el máximo y el mínimo en cada mitad.
- Regrese el máximo de los máximos y el mínimo de los mínimos.

Función maxmin(i, j)

1. Si $j - i \leq 1$ entonces
   1. regresa máx(a_i, a_j) y mín(a_i, a_j).
2. Si no entonces
   1. haz $(b, d) \leftarrow \text{maxmin}(i, \lfloor \frac{1}{2}(i+j) \rfloor)$
   2. haz $(c, e) \leftarrow \text{maxmin}(\lfloor \frac{1}{2}(i+j) \rfloor + 1, j)$
   3. regresa máx(b, c) y mín(d, e).

Análisis de tiempo de ejecución

- Sea $T(n)$ la cantidad de comparaciones hecha por maxmin(i, j) cuando $n = j - i + 1$.
- Supongamos que $n = 2^k$ para algún entero k.
- En este caso, cada uno de los subarreglos mide $n/2 = 2^{k-1}$ y por lo tanto tenemos que

$$T(n) = \begin{cases} 1 & \text{si } n = 2 \\ 2T(n/2) + 2 & \text{en otro caso.} \end{cases}$$

Sustitución

- Sustituyendo obtenemos que

$$\begin{aligned} T(n) &= 2T(n/2) + 2 \\ &= 2(2T(n/4) + 2) + 2 \\ &= 4T(n/4) + 4 + 2 \\ &= \ldots \\ &= 2^i T(n/2^i) + \sum_{j=1}^{i} 2^j \text{ para toda } 1 \leq j < k \\ &= 2^{k-1} T(2) + \sum_{j=1}^{k-1} 2^j \\ &= 2^{k-1} + (2^k - 2) \\ &= \frac{3}{2} n - 2. \end{aligned}$$