Divisão e Conquista

MERGE Sort

1948

Von Neumann

$$
\begin{array}{cccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7
\end{array}
$$

$$V = [\underbrace{2\ 4\ 6\ 8}_{p \quad q}\ \underbrace{1\ 3\ 5\ 7}_{r}]$$

MERGE (intercalação)

1 2 3 4 5 6 7 8

k

1 2 3 4 5 6 7 8

p q r

$L = 2\ 4\ 6\ 8\ \infty$

i

$R = 1\ 3\ 5\ 7\ \infty$

j

MERGE

$\Theta(n)$

entre p e r há
$n = r - p + 1$ elementos

$$
\begin{array}{cccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7
\end{array}
$$

V = [0 1 2 3 4 5 7 8]

0 2 4 7 | 1 3 5 8

2 7 | 0 4   1 5 | 3 8

7 | 2   4 | 0   5 | 1   8 | 3

Algorithm 2 MergeSort

1: **procedure** MERGESORT(V, p, r)
2: **if** $p < r$ **then**
3: $q = \lfloor \frac{p+r}{2} \rfloor$ — $\Theta(1)$
4: $MergeSort(V, p, q)$ — $n/2$ elementos — $T(n/2)$
5: $MergeSort(V, q+1, r)$ — $n/2$ elementos — $T(n/2)$
6: $Merge(V, p, q, r)$ — $\Theta(n)$
7: **end if**
8: **end procedure**

$$T(n) = \begin{cases} \Theta(1) & \text{se } n = 1 \\ \Theta(1) + T(n/2) + T(n/2) + \Theta(n) & \text{se } n > 1 \end{cases}$$

$\Downarrow$

Equação de Recorrência

$$T(n) = \begin{cases} \Theta(1) & \text{se } n = 1 \\ 2T(n/2) + \Theta(n). \end{cases}$$

$T(n) = \Theta(n \lg n).$

Algorithm 2 MergeSort

1: **procedure** MERGESORT(V, p, r)
2: **if** $p < r$ **then**
3: $q = \lfloor \frac{p+r}{2} \rfloor$
4: $MergeSort(V, p, q)$
5: $MergeSort(V, q+1, r)$
6: $Merge(V, p, q, r)$
7: **end if**
8: **end procedure**

Algorithm 1 Merge

```
 1: procedure MERGE(V, p, q, r)
 2:     n1 = q − p + 1
 3:     n2 = r − q
 4:     L[1..n1 + 1]
 5:     R[1..n2 + 1]
 6:     for i = 1; i ≤ n1; i++ do
 7:         L[i] = V[p + i − 1]   - θ(1)
 8:     end for
 9:     for j = 1; j ≤ n2; j++ do
10:         R[j] = V[q + j]   - θ(1)
11:     end for
12:     L[n1 + 1] = ∞
13:     R[n2 + 1] = ∞
14:     i = 1
15:     j = 1
16:     for k = p; k ≤ r; k++ do
17:         if L[i] ≤ R[j] then
18:             V[k] = L[i]
19:             i++
20:         else
21:             V[k] = R[j]
22:             j++
23:         end if
24:     end for
25: end procedure
```

Lines 2–5: $\Theta(1)$

Lines 6–8: $\Theta(n_1)$; lines 9–11: $\Theta(n_2)$; together: $\Theta(n)$.

Lines 12–15: $\Theta(1)$

Lines 18–19: $\Theta(1)$; lines 21–22: $\Theta(1)$; lines 17–23: $\Theta(1)$

n iterações $(r-p+1)$ c/ custo $\Theta(1)$ cada.
$n \cdot \Theta(1) = \Theta(n)$.

$V = [1\ 3\ 5\ 6\ \ 2\ 4\ 8\ 9]$

$$\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ p & & & q & & & & r \end{array}$$

$n_1 = 4$

$n_2 = 4$

	1	2	3	4	5
L	1 (i)	3	5	6	∞
R	2 (j)	4	8	9	∞

$$n = r - p + 1 = n_1 + n_2 = (q-p+1) + (r-q) = \cancel{q} - p + 1 + r - \cancel{q} = r - p + 1$$

Custo total: $T(n) = \Theta(1) + \Theta(n) + \Theta(1) + \Theta(n) = \Theta(n)$.

Análise

Seja $n = r - p + 1$. As linhas 2-5 e 12-15 executam em tempo constante. O laço for das linhas 6-8 executa $n_1 = q - p + 1$ vezes e o laço for das linhas 9-11 executa $n_2 = r - q$ vezes. Portanto, o custo do trecho das linhas 6-11 é $\cancel{q} - p + 1 + r - \cancel{q} = r - p + 1 = n = \Theta(n)$. O laço das linhas 16-24 repete $r - p + 1 = n$ vezes operações de tempo constante, portanto, seu custo é $\Theta(n)$. Assim o tempo total gasto é $\Theta(1) + \Theta(n) + \Theta(n) = \Theta(n)$.