CS 203 [Algorithms]

[Sorting Algorithms]

1. Insertion Sort :⟶

A[1..n] ← Input

⟶ Initialization step
⟶ Maintenance step
⟶ Termination step

for j = 2 to n ⟶ cost = c_1, # = n
Key = A[j] ← $c_2, n-1$
i = j-1 ← $c_3, n-1$ // insert key to correct position
while i > 0 and A[i] > Key ← $c_4, \sum_{j=2}^{n} t_j$
A[i+1] = A[i] ⟶ $c_5, \sum_{j=2}^{n} (t_j - 1)$
i = i-1 ⟶ $c_6, \sum_{j=2}^{n} (t_j - 1)$
A[i+1] = Key ⟶ $c_7(n-1)$

$$T(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4\left(\sum_{j=2}^{n} t_j\right) + c_5 \sum_{j=2}^{n} (t_j - 1) + c_6 \sum_{j=2}^{n} (t_j - 1) + c_7(n-1)$$

Best case – Already Sorted [∵ A[0] ≤ A[1] ⋯ ≤ A[n]

$$T(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4(n-1) + c_7(n-1)$$
$$= An + B$$
$$T(n) = \Theta(n)$$

Worst case :→ $A[1] \geq A[2] \geq A[3] \cdots A[n]$

$$T(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4 \sum_{j=2}^{n} c_j + c_5 \sum_{j=2}^{n} (c_j - 1) + c_6 \sum_{j=2}^{n} c_j - 1) + c_7(n-1)$$

$$\sum_{j=2}^{n} c_j - 1) = \frac{n(n-1)}{2}$$

$$\sum_{j=2}^{n} j = \frac{n(n+1)}{2} - 1$$

$$T(n) = c_1 n + (c_2 + c_3)(n-1) + c_4\left(\frac{n(n-1)}{2}\right) + (c_5 + c_6)\frac{n(n+1)}{2} + c_7(n-1)$$

$$= An^2 + Bn + C$$

$$= \Theta(n^2)$$

2. Merge Sort

~~func~~ Merge (A, p, q, r) {

⇒ $n_1 = q - p + 1$

⇒ $n_2 = r - q$

⇒ $L[1 : n_1 + 1]$, $R[1 : n_2 + 1]$

⇒ for (i = 1 to n_1) $\Theta(n_1)$

$L[i] = A[p + i - 1]$

⇒ for (j = 1 to n_2) $\Theta(n_2)$

$R[j] = A[q + j]$

⇒ $L[n_1 + 1] = \infty$

⇒ $R[n_2 + 1] = \infty$

$i = 1$, $j = 1$

for ($k = p :\rightarrow r$) $\Theta(n_1 + n_2)$

if ($L[i] \leq R[j]$)

$A[k] = L[i]$

$i++$

else

$A[k] = R[j]$

$j++$

$T(n) = \Theta(n_1 + n_2)$

$= A\Theta(n_1 + n_2) + B$

Divide $\longrightarrow \Theta(1)$

Conquer $\longrightarrow 2T\left(\frac{n}{2}\right)$

$$T(n) = 2T\left(\frac{n}{2}\right) + cn + \Theta(1)$$

Combine $\longrightarrow cn + c'$

Merge (A, p, q, r) [p ≤ r]

Sort

$$q = \left\lfloor \frac{p+r}{2} \right\rfloor$$

MergeSort (A, p, q)
MergeSort (A, q, r)
Merge (A, p, q, r)

$$T(n) = 2T\left(\frac{n}{2}\right) + cn$$

$n = 2^k$

$$T(2^k) = 2T(2^{k-1}) + c2^k$$
$$T(2^{k-1}) = 2T(2^{k-2}) + c2^{k-1}$$
$$2T(2^{k-1}) = 4T(2^{k-2}) + c2^k$$

$$T(2^k) - 2T(2^{k-1}) = 2T(2^{k-1}) - 4T(2^{k-2})$$
$$T(2^k) - 4T(2^{k-1}) + 4T(2^{k-2}) = 0$$
$$S(k) - 4S(k-1) + 4S(k-2) = 0$$

$S(k) = x^k$

$$x^k - 4x^{k-1} + 4x^{k-2} = 0$$
$$x^2 - 4x + 4 = 0$$
$$(x-2)^2 = 0 \Rightarrow x = 2, 2$$

$$S(k) = c_1 2^k + c_2 k 2^k$$

$$T(n) = c_1 n + c_2 n \log n$$

$$T(n) = \Theta(n \log n)$$