

Merge Sort 3 Split Recurrence Relation

- 1) if the array has 0 or 1 elements, it's sorted. $T(1) = 1$
- 2) Split the array into 3 approximately equal sized thirds 1
- 3) sort each third recursively (using mergesort). $3T(\frac{n}{3})$
- 4) merge the sorted thirds to produce one sorted result. n

$$T(n) = \begin{cases} 1 & \text{if } n \leq 1 \\ 3T(\frac{n}{3}) & \text{if } n > 1 \end{cases}$$

Solve by Substitution

$$T(n) = 3T(\frac{n}{3}) + n$$

$$= 3(3T(\frac{n}{9}) + \frac{n}{3}) + n$$

$$= 9T(\frac{n}{9}) + 3n$$

$$= 18T(\frac{n}{18}) + 4n$$

$$= 3^i T(\frac{n}{3^i}) + in \quad \text{for } i = \lg n$$

$$= nT(1) + n \lg n = n + n \lg n \in \Theta(n \lg n)$$