Time Complexity of Small Large Merge Sort Algorithm

20BRS1193

$$T(n) = T(2) + T(n-2) + n \qquad - ①$$

sub n with $n-2$:

$$T(n-2) = T(2) + T(n-4) + n-2 \qquad - ②$$

sub $T(n-2)$ with ② in ①

$$T(n) = T(2) + T(2) + \{ T(n-4) + n-2 + n$$

$$T(n) = 2T(2) + T(n-4) + 2n - 2 \qquad - ③$$

Sub $n-4$ in place of in ①

$$T(n-4) = T(2) + T(n-6) + n-4 \qquad ④$$

So ③ becomes —

$$T(n) = 3T(2) + T(n-6) + 3n - 6$$

$$\vdots$$

$$4T(2) + T(n-8) + 4n - 8$$
$$5T(2) + T(n-10) + 5n - 10$$
$$6T(2) + T(n-12) + 6n - 12$$

$$\vdots$$

$$i \, T(2) + T(n-2i) + in - 2i$$

$$T(1) = 1$$

Let $n - 2i = 1$

$$n = 1 + 2i$$

$$\log n = \log(1 + 2i)$$

$$i = \frac{n-1}{2}$$

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

$$T(2) = 4$$

$$= \left(\frac{n-1}{2}\right)4 + 1 + \frac{n^2 - n}{2} - \frac{2n + 2}{2}$$

$$= \frac{4n - 4 + 2 + n^2 - n - 2n + 2}{2}$$

$$\boxed{T(n) = \frac{n^2 + n}{2}}$$