

2. Considere o algoritmo xpto. Indique a complexidade do mesmo, no pior caso, em função de n . Indique a equação de recorrência e resolva-a.

```
public static void xpto(int[] v, int n) {
    if (n <= 1) return;
    n=n/2;
    for(i=0; i<n; i=i+1)
        v[i] = v[2i] + v[2i +1]
    xpto(v, n)
}
```

$$\begin{aligned} T(n) &= T(n/2) + O(n) + O(1) = \\ &= T(n) = T(n/2) + O(n) \end{aligned}$$

Método de substituições

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

$$T(n) = T\left(\frac{n}{4}\right) + \frac{n}{2} + n =$$

$$T(n) = T\left(\frac{n}{8}\right) + \frac{n}{4} + \frac{n}{2} + n =$$

$$T(n) = T\left(\frac{n}{16}\right) + \frac{n}{8} + \frac{n}{4} + \frac{n}{2} + n =$$

$$T(n) = \frac{(2^k - 1)}{2^{k-1}} n + T(1), \quad k = \log_2(n)$$

$$T(n) = \frac{(2^k - 1)}{2^k 2^{-1}} n =$$

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

$$T(n) = 2n - 2 =$$

$$= T(n) = n$$

A complexidade temporal é de $O(n)$