

# Finding of Power of an Element

$$a \rightarrow 2$$

$$n \rightarrow 10$$

$$\text{Output} = a^n = 2^{10} = \underline{\underline{1024}}$$

$$\hookrightarrow \underline{\underline{n \geq 1}} \leftarrow$$

$$\underline{\underline{n < 1}} \leftarrow$$

$$\begin{cases} 2^1 = 2 & 5^1 = 5 \\ 4^1 = 4 \end{cases}$$

Divide & conquer

Small problem  $\rightarrow n = 1$

$\hookrightarrow$  return solution  $\hookrightarrow$  return a

big problem  $\rightarrow n > 1$

$\hookrightarrow$  Divide & conquer  $a^n = a^{n/2} \times a^{n/2}$

$$\underline{\underline{n = 64}}$$

$$\underline{\underline{a = 2}}$$

$$\underline{\underline{n // 2}}$$

$$b = 2^{32}$$

$$\hookrightarrow \underline{b * b}$$

Recursive Tree

$n \rightarrow \text{even}$

$$\hookrightarrow 2^{10} = \underline{\underline{1024}}$$

$n \rightarrow \text{odd} \rightarrow 2^{11}$

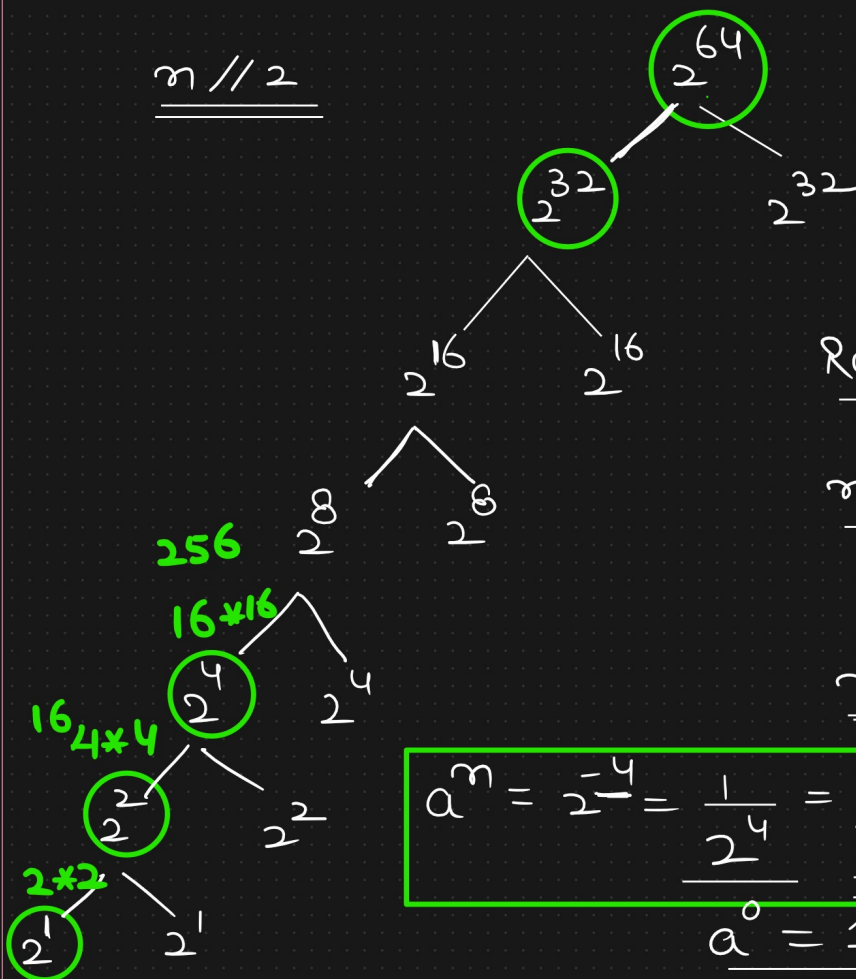
$$2^{10} \times \frac{2}{a}$$

$$n = -4$$

$$a = 2$$

$$a^n = 2^{-4} = \frac{1}{2^4} = \underline{\underline{\frac{1}{16}}}$$

$$\underline{\underline{a^0 = 1}}$$



$T(n)$  — findPower(a, n):  $n \geq 1$

$n = 0$   
 $\hookrightarrow$  return 1

$n < 1$   
 $\hookrightarrow$   $a = 1/a$

$n = -n$

return findPower(a, n)

$\left\{ \begin{array}{l} \text{if } n == 1: \rightarrow \text{small Problem} \\ \text{return } a \\ \text{else: } \rightarrow \text{Big Problem} \\ \text{mid} = n // 2 \quad \hookrightarrow \text{Divide \&} \\ \text{b} = \text{findPower}(a, \text{mid}) \quad \text{Conquer} \\ \text{result} = b * b \quad \text{Approach} \\ \text{if } n \% 2 == 0 \rightarrow \text{Even} \\ \text{return result} \\ \text{else: } \rightarrow \text{Odd} \\ \text{return result} * a \end{array} \right.$

Recurrence Relation

$$\hookrightarrow T(n) = T(n/2) + c$$

$\hookrightarrow$  Master's Theorem or Substitution Method

Master's Theorem

$$a = 1 \quad k = 0$$

$$b = 2 \quad p = 0$$

$$\log_b a = \log_2 1 = 0$$

$$\log_b a = k = 0 \text{ — } \underline{\underline{\text{case 2}}}$$

$$p > -1 \Rightarrow p = 0$$

$$\underline{\underline{\Theta(\log n)}}$$

Substitution Method  $\rightarrow$  Recursive Term

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

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

$$= T\left(\frac{n}{2^3}\right) + c + c + c$$

$$\frac{n}{2^k} = 1 \quad \left. \begin{array}{l} \downarrow \\ k \text{ times} \end{array} \right\}$$

$$\underline{\underline{k = \log_2 n}}$$

$$= T\left(\frac{n}{2^k}\right) + k \cdot c$$

$$= T\left(\frac{n}{2^{\log_2 n}}\right) + k \cdot c$$

$$= T\left(\frac{n}{n^{\log_2 2}}\right) + k \cdot c$$

$$= T(1) + k \cdot c$$

$$= 1 + \log_2 n \cdot c$$

$$\Rightarrow \underline{\underline{O(\log_2 n)}}$$