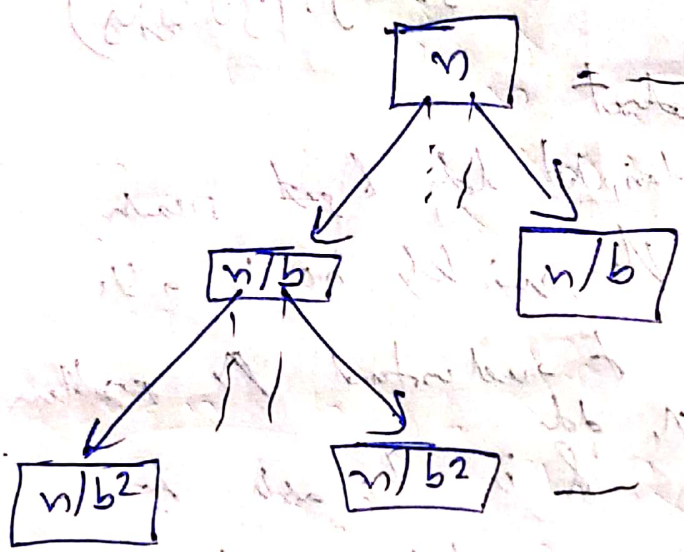
$$T(n) = aT\left(\frac{n}{b}\right) + O(n^d)$$

$$T(n) = \begin{cases} O(n^d) & d > \log_b a \\ O(n^d \log n) & d = \log_b a \\ O(n^{\log_b a}) & d < \log_b a \end{cases}$$

b → size of the sub-problem

a → number of sub-problems



	Size	#	Complexity
	n	1	$O(n^d)$
	$\frac{n}{b}$	a	$a\,O\left(\left(\frac{n}{b}\right)^d\right)$
	$\frac{n}{b^2}$	a^2	$a^2\,O\left(\left(\frac{n}{b^2}\right)^d\right)$
i^{th} level	$\frac{n}{b^i}$	a^i	$\left(\frac{a}{b^d}\right)^i O(n^d)$
1 1 1	$1 = \frac{n}{b^H}$	a^H	$a^{\log_b n} = n^{\log_b a}$

$$\frac{n}{b^H} = 1 \Rightarrow \boxed{H = \log_b n}$$

{ Prove by using value of H }

Total time complexity

$$= O(n^d)\left[1 + \frac{a}{b^d} + \left(\frac{a}{b^d}\right)^2 + \cdots\right]$$

$$= O(n^d) \times \frac{\left(1 - \left(\frac{a}{b^d}\right)^4\right)}{1 - \frac{a}{b^d}}$$

* In a G.P., first term dominates when ~~r > 1~~ $r < 1$

$\Rightarrow \frac{a}{b^d} < 1 \Rightarrow d > \log_b a$

$\Rightarrow T(n) = O(n^d)$

* Last term dominates when $r > 1$

$\Rightarrow T(n) = n^{\log_b a}$

* when $r = 1$, the formula doesn't holds.. All terms are equal so →

$$T(n) = \sum_{0}^{4} O(n^d)$$

$$= (1 + \log_b n)\, O(n^d)$$

$$\Rightarrow \boxed{T(n) = O(n^d \log n)}$$