Árvore de Recursão

$$T(n) = \begin{cases} 1 & ; \; n = 1 \\ 2T(n/2) + n & ; \; n > 1 \end{cases}$$

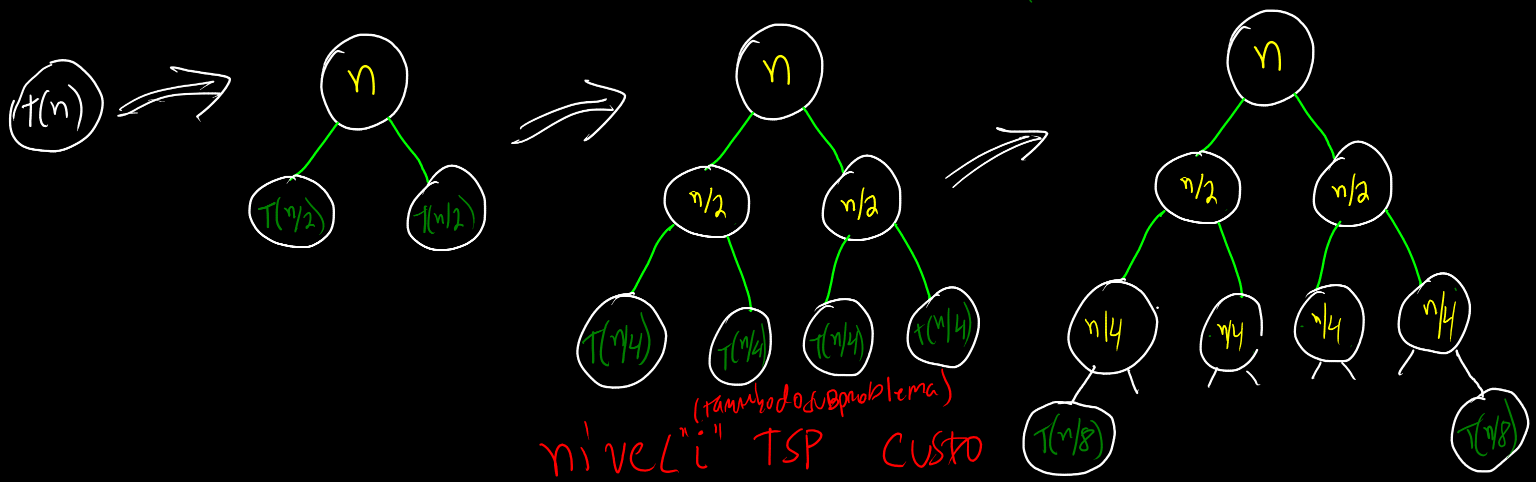
custo do caso base

qtos subproblemas

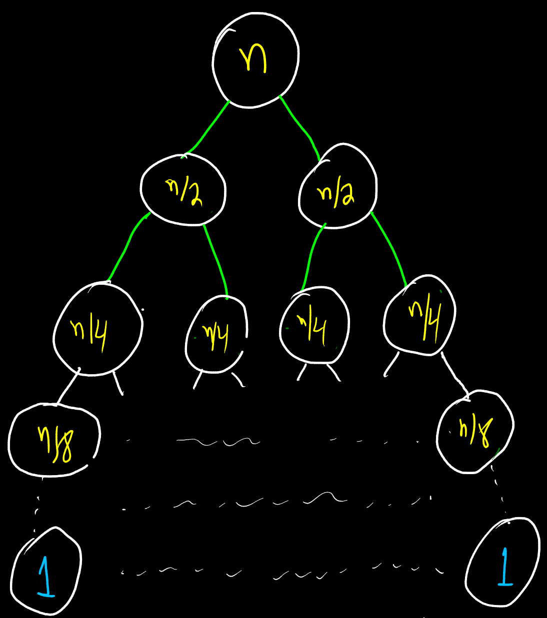
tx de redução do tamanho do problema

custo do subproblema

n^2



nível "i"	TSP (tamanho do subproblema)	custo
0	$n = \frac{n}{2^0}$	n
1	$n/2 = \frac{n}{2^1}$	$n/2 + n/2 = n$
2	$n/4 = \frac{n}{2^2}$	$n/4 + n/4 + n/4 + n/4 = n$
3	$n/8 = \frac{n}{2^3}$	$8(\frac{n}{8}) = n$
⋮	⋮	⋮
k	$1 = \frac{n}{2^k}$	?



olhando na tabela:
em nível i
custo$(i) = n$
TSP$(i) = n/2^i$

$$1 = \frac{n}{2^k}$$
$$2^k = n$$
$$lg(2^k) = lg(n)$$
$$k \, \cancel{lg(2)}^{\,1} = lg(n)$$
$$k = lg(n).$$

Supondo que a árvore está completa, e sabendo que a árvore é binária, há 2^i nós em um nível i. Portanto, no último nível (k) temos $2^k = 2^{lg(n)} = n^{lg(2)} = n^1 = n$ nós. Portanto, o custo total do último nível é $n \cdot 1 = \Theta(n)$.

Sabemos que o custo de cada nível intermediário é n e que os níveis intermediários vão de $0..k-1 = 0..lg(n)-1$. Portanto o custo total da árvore é:

$$T(n) = \sum_{i=0}^{lg(n)-1} n + \Theta(n)$$

custo dos níveis $0..k-1$.

custo do último nível (k)

$$= n \sum_{i=0}^{lg(n)-1} 1 + \Theta(n)$$
$$= n \cdot (lg(n) - 1) + \Theta(n) = n lg n - n + \Theta(n)$$
$$= \Theta(n lg n).$$

Árvore de Recursão

$$T(n) = \begin{cases} 1 & ; \ n = 1 \\ 2T(n/2) + 1 & ; \ n \geq 1 \end{cases}$$

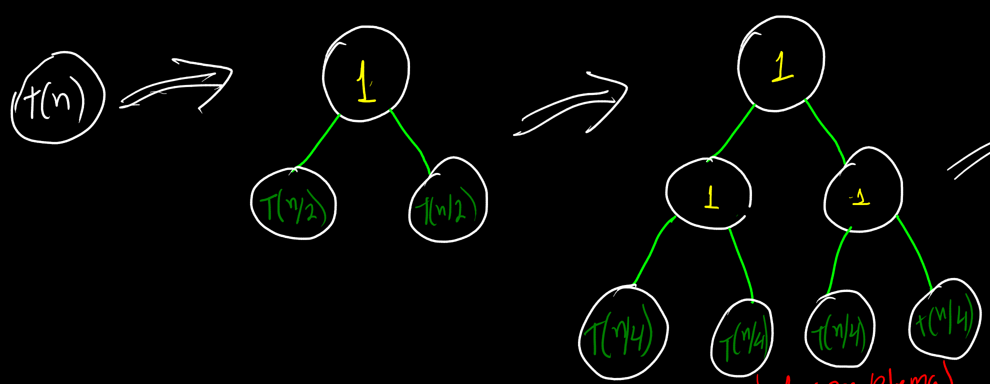
n^2

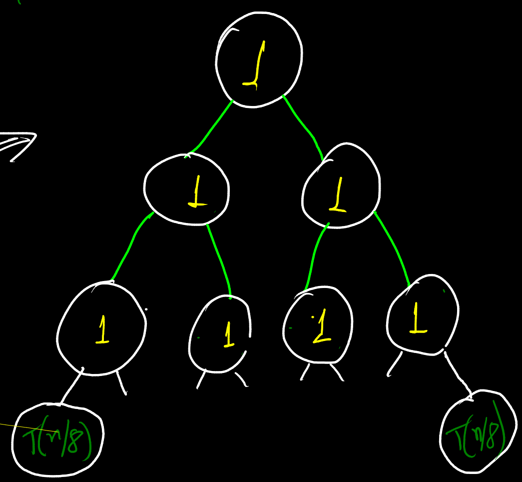
custo do caso base

qtos subproblemas

tx de redução do tamanho do problema

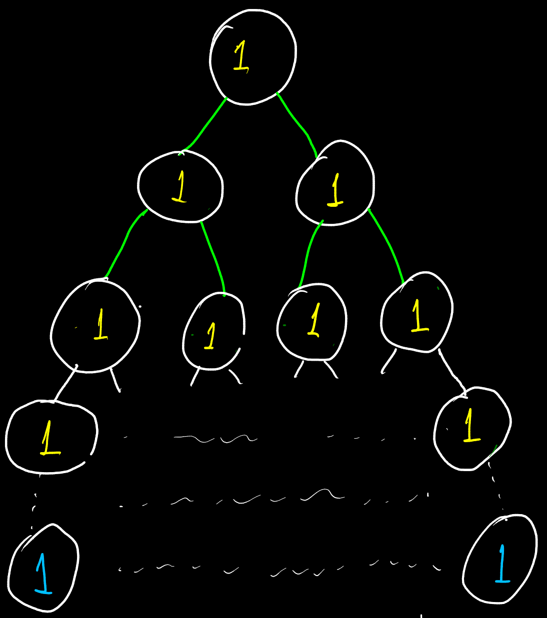
custo do subproblema





(tamanho do subproblema)

nível "i"	TSP	custo
0	$n = \frac{n}{2^0}$	$1 = 2^0$
1	$n/2 = \frac{n}{2^1}$	$2 = 2^1$
2	$n/4 = \frac{n}{2^2}$	$4 = 2^2$
3	$n/8 = \frac{n}{2^3}$	$8 = 2^3$
⋮	⋮	⋮
k	$1 = \frac{n}{2^k}$	?



olhando na tabela:
em nível i
$custo(i) = 2^i$
$TSP(i) = n/2^i$

$$1 = \frac{n}{2^k}$$
$$2^k = n$$
$$lg(2^k) = lg(n)$$
$$k \cancel{lg(2)} = lg(n)$$
$$k = lg(n).$$

Supondo que a árvore está completa, e sabendo que a árvore é binária, há 2^i nós em um nível i. Portanto, no último nível (k) temos $2^k = 2^{lg(n)} = n^{lg(2)} = n^1 = n$ nós. Portanto, o custo total do último nível é $n \cdot 1 = \Theta(n)$.

O custo de cada nível intermediário é 2^i. Sabemos que os níveis intermediários vão de $0..k-1 = 0..lg(n)-1$. Portanto o custo total da árvore é:

$$T(n) = \sum_{i=0}^{lg(n)-1} 2^i + \Theta(n)$$

custo dos níveis $0..k-1$

custo do último nível (k)

$$= 2^{lg(n)-1+1} - 1 + \Theta(n)$$
$$= \underbrace{2^{lg(n)}}_{n} - 1 + \Theta(n) = n - 1 + \Theta(n) = \Theta(n).$$

$$\boxed{\sum_{i=0}^{j} x^i = \frac{x^{j+1} - 1}{x - 1}}$$