EN 2023

① $T(n) = a\,T\left(\frac{n}{b}\right) + n^c = 1\,T\left(\frac{n}{3}\right) + 1$

$a=1$ (1 chamada recursiva em cada subproblema)

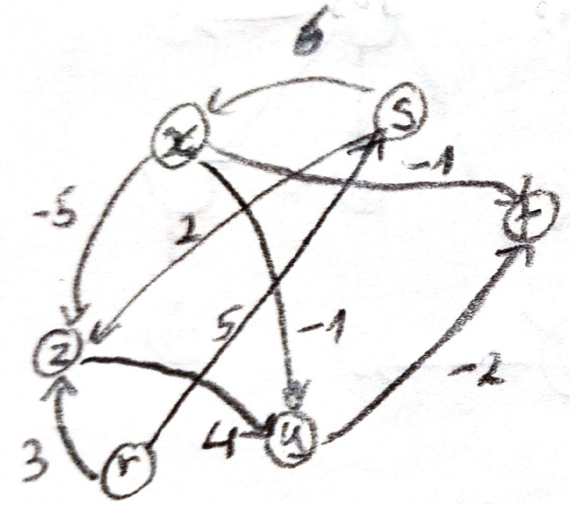
$b=3$ (O tamanho do input é dividido por 3 em cada subproblema)

$c=0$ (O trabalho não-recursivo de cada subproblema é constante

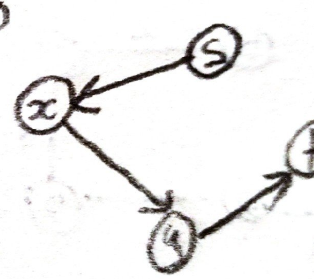
$\log_b a = \log_3 1 = 0 = c \Rightarrow \Theta(n^0 \log n) = \Theta(\log n)$

②

ⓐ



node	s	x	y	z	r	t
deg	1	1	2	3	0	2
	0	0	1	2		1
			0	1		0
				0		

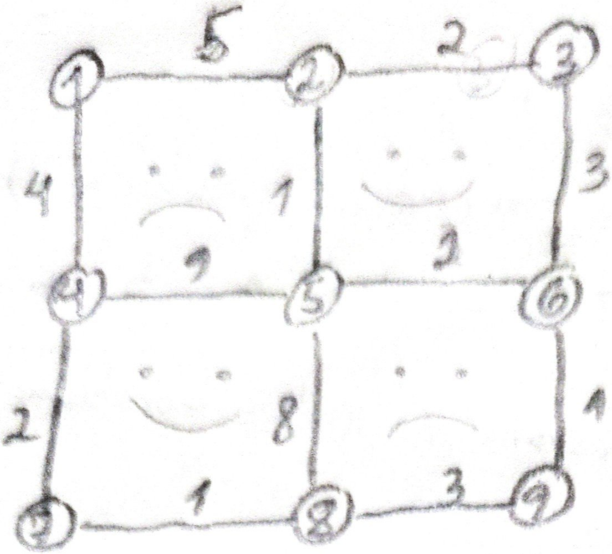


topological order:	r	s	x	z	y	t
d:	∞	0	∞	∞	∞	∞
			6	2	5	5
				1		3

ⓑ O caminho mais curto até cada edge v é dado pelo mínimo de $d(u)+w(u,v)$ para cada antecessor u de v.

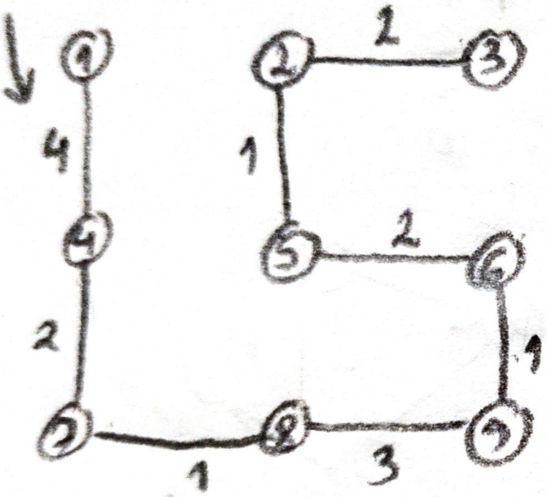
Como tal, a solução ótima para o problema de encontrar $d(v)$ é construída a partir das soluções ótimas de subproblemas (encontrar $d(u)$ para os antecessores

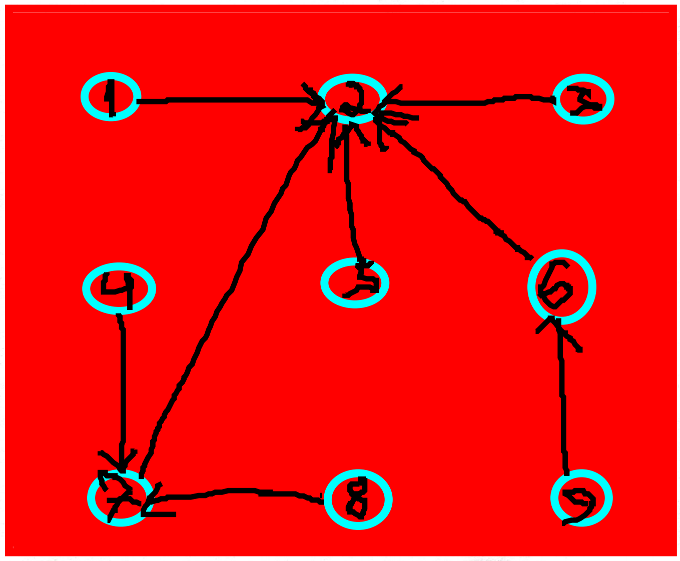
③



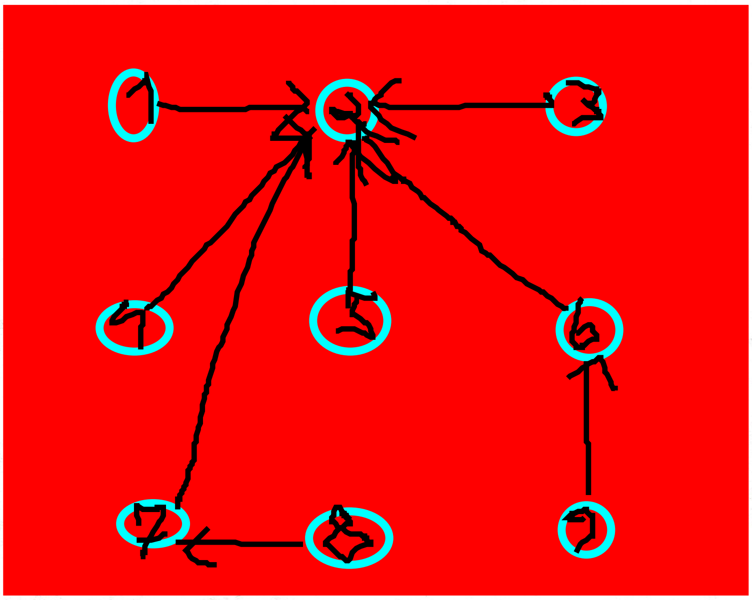
ⓐ

última





ⓑ



(4)

$$C(s,i,j) = \begin{cases} 0, & s_i \ldots s_j \text{ é palíndromo} \\ \min\limits_{i \le K < j} \{ C(s,i,K) + 1 + C(s,K+1,j) \}, & \text{else} \end{cases}$$

```
function C(s, n):                    n=|s|
    dp[x][x] = [0 for x<n]   //se j=i, min cts é 0
    for L in 0,...n:
        for i in 0,..., n-L:
            j = i+L
            if (Palindrome(s[i:j]) = True):
                dp[i,j] = 0
            else:
                dp[i,j] = ∞
                for K in i,...,j-1:
                    ct = dp[i,j] + 1 + dp[K+1, j]
                    if ct < dp[i,j]:
                        dp[i,j] = ct
    return dp[0,n]
```

Complexidade Temporal:

$O(n^2)$ estados dp[x][y]

Para cada estado:

Tendo que $O(g(n))$ é a complexidade de Palindrome()

Loop de i até j-1, no pior caso $O(n)$

$$O(n^2)\,O(n + g(n)) = \begin{cases} O(n^3) & \text{se } g(n) \le n \\ O(n^2 g(n)) & \text{se } g(n) > n \end{cases}$$

Complexidade Espacial: $O(n^2)$ pois há uma tabela com n^2 estados