

EN 2023

① $T(n) = aT(\frac{n}{b}) + n^c = 1T(\frac{n}{3}) + 1$

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

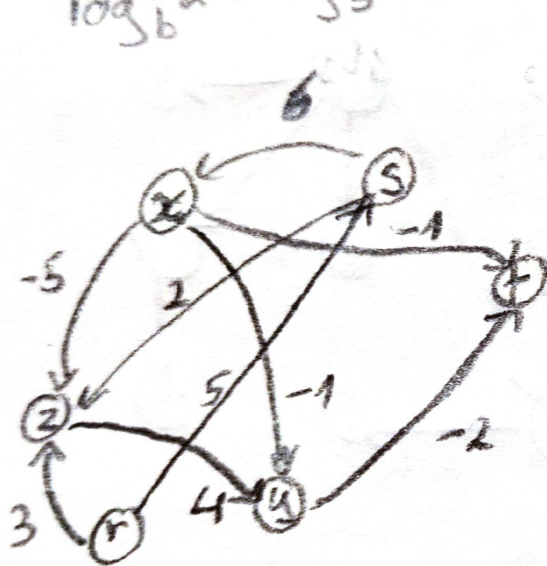
$b=3$ (O tamanho do input é dividido por 3 em cada subproblema)

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

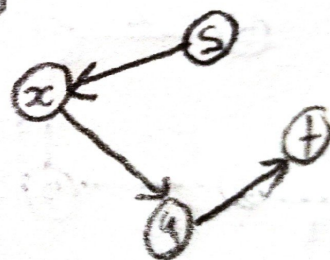
$\log_b a = \log_3 1 = 0 = c \Rightarrow \Theta(n^c \log n) = \Theta(\log n)$

②

③



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

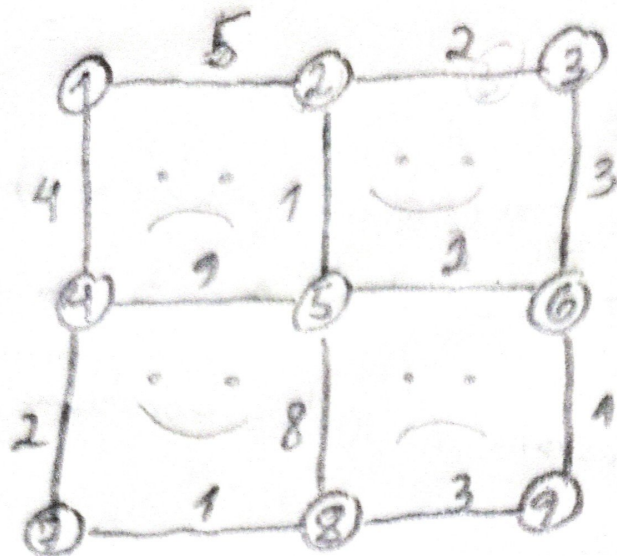


topological order: r s x z y t
d: ∞ 0 ∞ ∞ ∞ ∞
6 2 5 5
1 3

④ O caminho mais curto até cada edge v é dado pelo mínimo de $d(u) + w(u, v)$ para cada antecessor u de v .

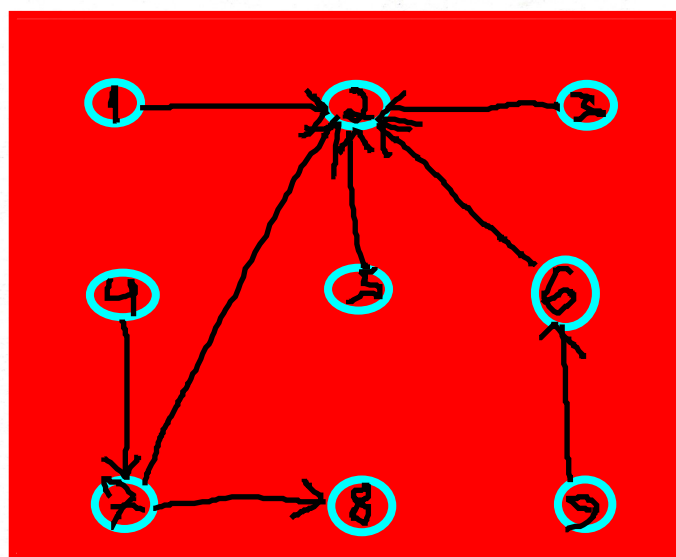
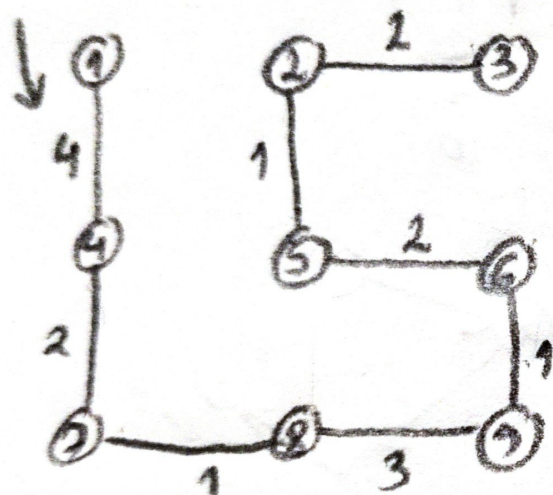
Como tal, a solução ótima para o problema de encontrar $d(v)$ é construída a partir das soluções ótimas de subproblemas (encontrar $d(u)$ para os antecessores

③

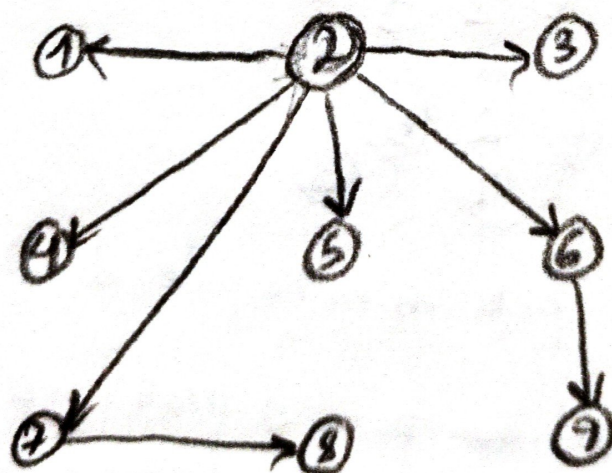


②

Ultima



⑥



④

$$C(s, i, j) = \begin{cases} 0, & \text{se } s_i \dots s_j \text{ é palíndromo} \\ \min_{i \leq k < j} \{ C(s, i, k) + 1 + C(s, k+1, j) \}, & \text{else} \end{cases}$$

function $C(s, n)$:

$n = |s|$

$dp[x][x] = 0$ for $x < n$ // se $j = i$, min cts é 0

for L in $0 \dots n$:

for i in $0 \dots n-L$:

$j = i + L + 1$ // $(s[i:j]) = \text{True}$

if $(\text{Palindrome}(s[i:j]) = \text{True})$:

$dp[i, j] = 0$

else:

$dp[i, j] = \infty$

for k in $i \dots j-1$:

$ct = dp[i, j] + 1 + dp[k+1, j]$

if $ct < dp[i, j]$:

return $dp[0, n]$ $dp[i, j] = ct$

Complexidade Temporal: $O(n^2 g(n))$ onde $g(n)$ é a complexidade de Palindrome()

Para cada estado:

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

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

$$O(n^2) O(n + g(n)) =$$

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

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

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