

CHECKLIST

fez ☐ incompleto ☐ não entendi ☐

- ☐ 1. construir matriz dfa[][]
- ☐ 2. simular KMP
- ☐ 3. escrever expressão regular
- ☐ 4. construir autômato finito
- ☐ 5. ver se palavra é aceita no autômato, mostrar um caminho
- ☐ 6. árvore do código de Huffman
- ☐ 7. usando o ex. 6. mostrar a codificação e quantos bits usou
- ☐ 8. decodificar com a árvore de Huffman

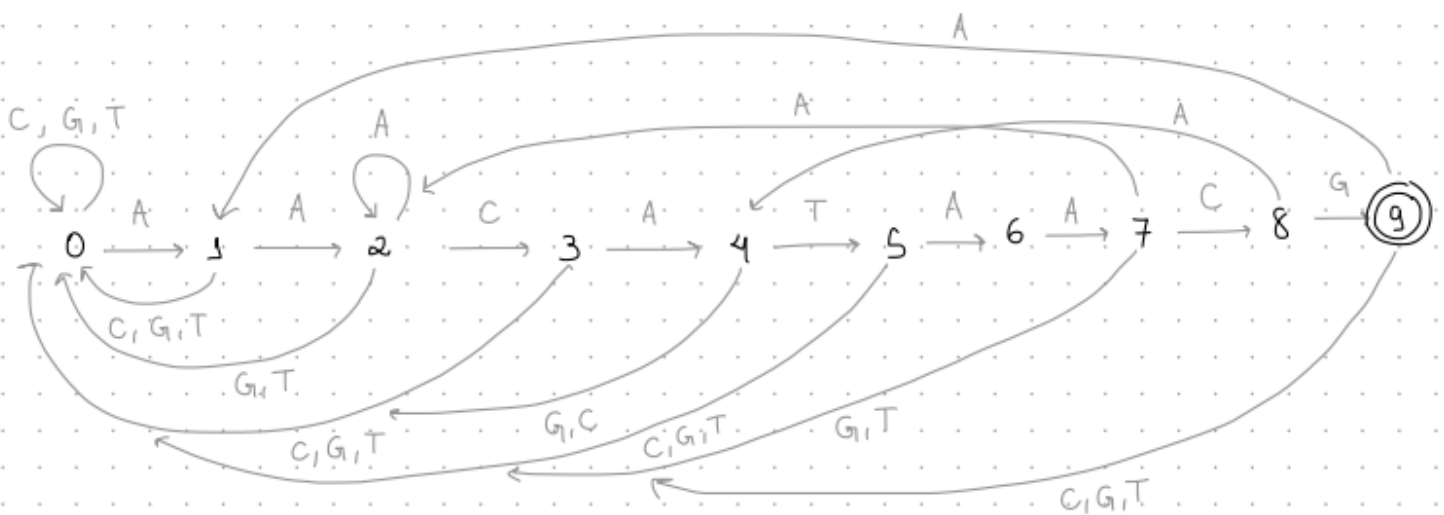
1. Construa a matriz $dfa[][]$ para os seguintes padrões do alfabeto {A, C, G, T}:

- AACATAACG
- TTTTTT
- ATAATAAAT

	0	1	2	3	4	5	6	7	8
	A	A	C	A	T	A	A	C	G
A	1	2	2	4	2	6	7	2	4
C	0	0	3	0	0	0	0	8	0
G	0	0	0	0	0	0	0	0	9
T	0	0	0	0	5	0	0	0	0

	0	1	2	3	4	5
	T	T	T	T	T	T
T	1	2	3	4	5	6

	0	1	2	3	4	5	6	7	8
	A	T	A	A	T	A	A	A	T
A	1	1	3	4	1	6	7	8	1
T	0	2	0	2	5	0	2	5	9



2. Simule a execução do algoritmo KMP, como visto em sala de aula, para cada um dos padrões no texto abaixo:

ATTTATAATAATAATAAATTTTTTTTAACATAACATAACG

ATTTATAATAATAATAAATTTTTTTTAACATAACATAACG

para AACATAACG

ATTTATAATAATAATAAATTTTTTTTAACATAACATAACG
AACATAACG

ATTTATAATAATAATAAATTTTTTTTAACATAACATAACG
AACATAACG

ATTTATAATAATAATAAATTTTTTTTAACATAACATAACG
AACATAACG

ATTTATAATAATAATAAATTTTTTTTAACATAACATAACG
AACATAACG

ATTTATAATAATAATAAATTTTTTTTAACATAACATAACG
AACATAACG

ATTTATAATAATAATAAATTTTTTTTAACATAACATAACG
AACATAACG

ATTTATAATAATAATAAATTTTTTTTAACATAACATAACG
AACATAACG

ATTTATAATAATAATAAATTTTTTTTAACATAACATAACG
AACATAACG

ATTTATAATAATAATAAATTTTTTTTAACATAACATAACG
AACATAACG

ATTTATAATAAATAATAAATTTTTTTTAAACATAACATAACG
AACATAACG

ATTTATAATAAATAATAAATTTTTTTTAAACATAACATAACG
AACATAACG

ATTTATAATAAATAAATAAATTTTTTTTAAACATAACATAACG
AACATAACG

ATTTATAATAAATAAATAAATTTTTTTTAAACATAACATAACG
AACATAACG

ATTTATAATAAATAAATAAATTTTTTTTAAACATAACATAACG
AACATAACG

depois de passar por todas as letras T...

ATTTATAATAAATAAATAAATTTTTTTTAAACATAACATAACG
AACATAACG

ATTTATAATAAATAAATAAATTTTTTTTAAACATAACATAACG
AACATAACG

ATTTATAATAAATAAATAAATTTTTTTTAAACATAACATAACG
AACATAACG

encontramos o padrão no texto

ATTTATAATAAATAATAAATTTTTTTTAAACATAACATAACG
para TTTTTT

ATTTATAATAAATAATAAATTTTTTTTAAACATAACATAACG
TTTTTTT

ATTTATAATAAATAATAAATTTTTTTTAAACATAACATAACG
TTTTTTT

ATTTATAATAATAATAAATTTTTTTTAAACATAACATAACG
TTTTTTT

ATTTATAATAAATAATAAATTTTTTTTAAACATAACATAACG
TTTTTTT

ATTTATAATAATAATAAATTTTTTTTAAACATAACATAACG
TTTTTTT

ATTTATAATAAATAATAAATTTTTTTTAAACATAACATAACG
TTTTTTT

ATTTATAATAAATAATAAATTTTTTTTAAACATAACATAACG
TTTTTTT

ATTTATAATAAATAATAAATTTTTTTTAAACATAACATAACG
TTTTTTT

ATTTATAATAAATAATAAATTTTTTTTAAACATAACATAACG
TTTTTTT

ATTTATAATAAATAATAAATTTTTTTTAAACATAACATAACG
TTTTTTT

ATTTATAATAAATAATAAATTTTTTTTAAACATAACATAACG
TTTTTTT

ATTTATAATAAATAATAAATTTTTTTTAAACATAACATAACG
TTTTTT

ATTTATAATAAATAATAAATTTTTTTTAAACATAACATAACG
TTTTTT

achamos o padrão no texto

ATTTATAATAAATAATAAATTTTTTTTAAACATAACATAACG

para ATAATAAAT

ATTTATAATAAATAATAAATTTTTTTTAAACATAACATAACG
ATAATAAAT

ATTTATAATAAATAATAAATTTTTTTTAAACATAACATAACG
ATAATAAAT

ATTTATAATAAATAATAAATTTTTTTTAAACATAACATAACG
ATAATAAAT

achamos o padrão no texto

3. Escreva uma expressão regular para as seguintes linguagens:

- números binários ímpares; número par termina em 0 e ímpar em 1
- nomes de variáveis em C; não começa com número
- CEPs do estado de São Paulo; 01000 - 000 a 19999 - 999

1. $(0|1)^*1$

2. $([a-z]|[A-Z])([a-z]|[0-9]|[A-Z])^*$

3. $(0|1)([1-9])([0-9])\{3\} \setminus - ([0-9])\{3\}$

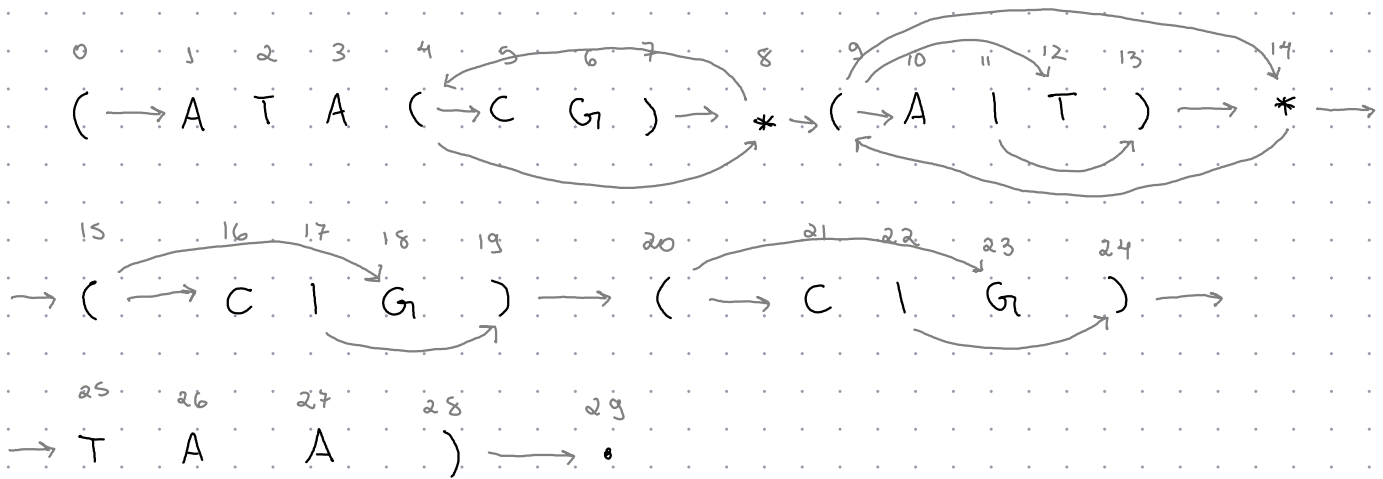
• parentesis (

• tem arco para o próximo caractere

• para 1 ; de 1 vai para o fecha parentesis

4. Construa o autômato finito não determinístico para a expressão regular

$(ATA(CG)^*(A|T)^*(C|G)(C|G)TAA)$



5. Para cada palavra abaixo, verifique se é aceita pela expressão regular acima, e, caso seja, mostre um caminho no autômato:

- ATACGTTTGGTAA
- ATACGTAA
- ATAAAATGGTAA

para $\overset{\cdot}{A}\overset{\cdot}{T}\overset{\cdot}{A}\overset{\cdot}{C}\overset{\cdot}{G}\overset{\cdot}{T}\overset{\cdot}{T}\overset{\cdot}{T}\overset{\cdot}{G}\overset{\cdot}{G}\overset{\cdot}{T}\overset{\cdot}{A}\overset{\cdot}{A}$

caminho: 1, 2, 3, 5, 6, 12, 12, 12, 18, 23, 25, 26, 27

$\{0\} \rightarrow \{0, \underline{1}\} \quad \text{lê } A$

$\{2\} \rightarrow \{\underline{2}\} \quad \text{lê } T$

$\{3\} \rightarrow \{\underline{3}\} \quad \text{lê } A$

$\{4\} \rightarrow \{4, \underline{5}, 8, 9, 10, 12, 14, 15, \underline{16}, 18\} \quad \text{lê } C$

$\{6, 17\} \rightarrow \{\underline{6}, 17, 19, 20, 21, \underline{23}\} \quad \text{lê } G$

$\{7, 24\} \rightarrow \{7, 24, 8, 9, 10, \underline{12}, 14, 15, 16, 18, \underline{25}\} \quad \text{lê } T$

$\{13, 26\} \rightarrow \{13, 26, 14, 9, 10, \underline{12}, 15, 16, 18\} \quad \text{lê } T$

$\{13\} \rightarrow \{13, 14, 9, 10, \underline{12}, 15, 16, 18\} \quad \text{lê } T$

$\{13\} \rightarrow \{13, 14, 9, 10, 12, 15, 16, \underline{18}\} \quad \text{lê } G$

$\{19\} \rightarrow \{19, 20, 21, \underline{23}\} \quad \text{lê } G$

$\{24\} \rightarrow \{24, \underline{25}\} \quad \text{lê } T$

$\{26\} \rightarrow \{\underline{26}\} \quad \text{lê } A$

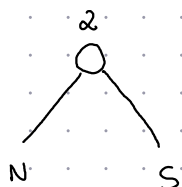
$\{27\} \rightarrow \{\underline{27}\} \quad \text{lê } A$

$\{28\} \rightarrow \{28, \underline{29}\} \quad \text{final}$

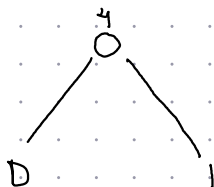
6. Mostre uma árvore construída com base nos códigos de Huffman para as seguintes frequências de uso das letras abaixo:

D: 2	E: 5	I: 2
N: 1	O: 3	P: 3
R: 4	S: 1	

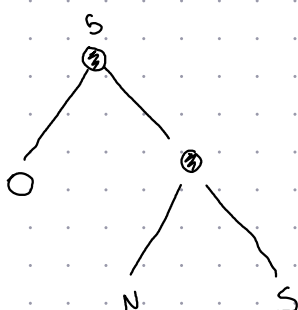
N e S = NS



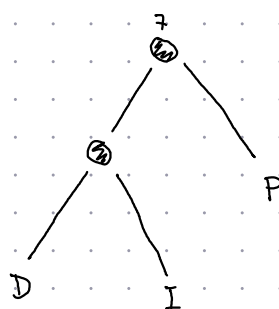
D e I = DI



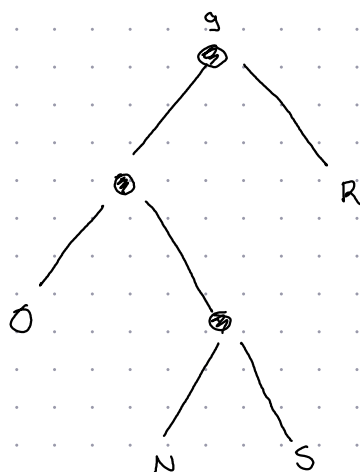
NS e O = ONS



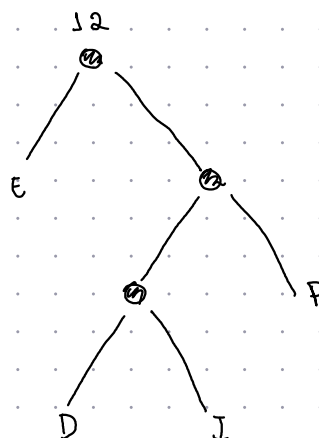
DI e P = DIP



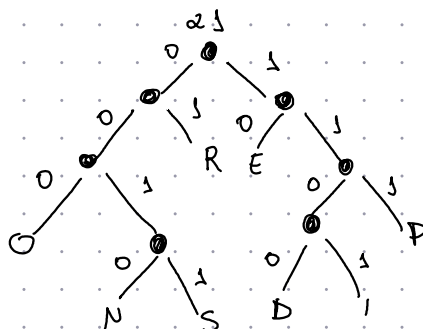
ONS e R = RONS



DIP e E = DIPE

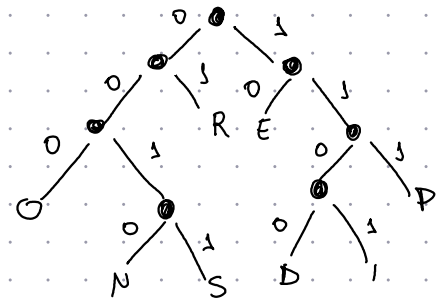
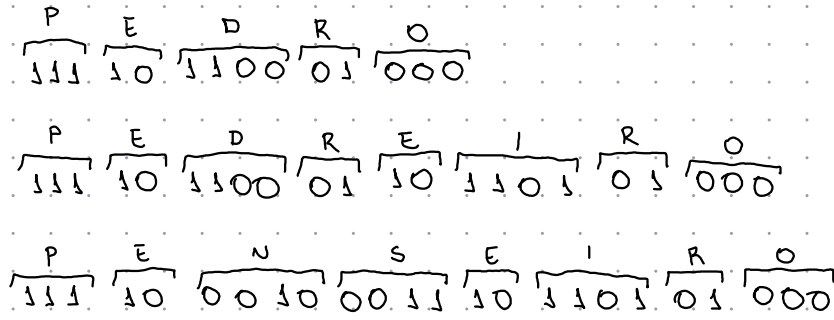


DIPE e RONS

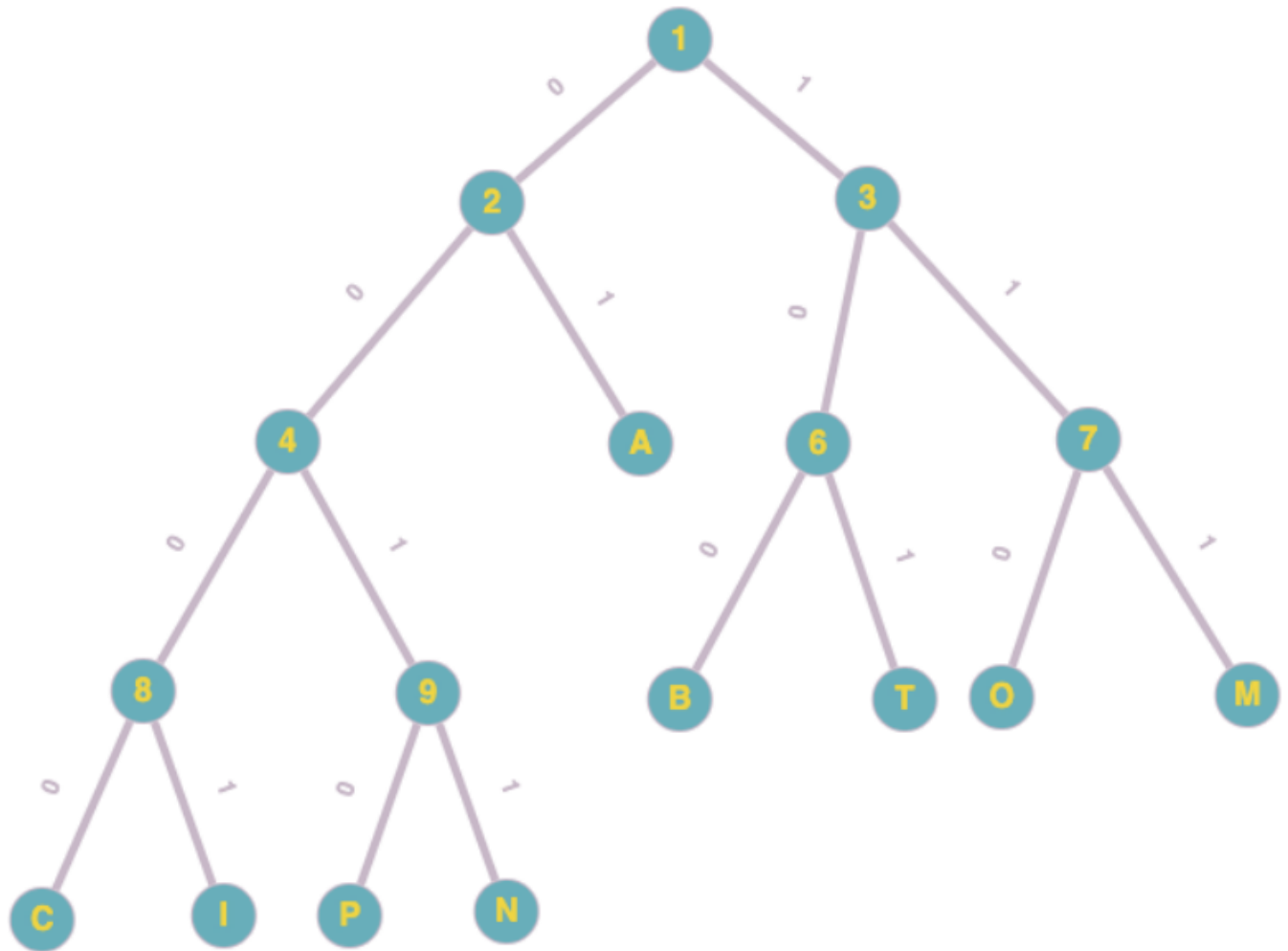


7. Usando a árvore construída no item anterior, mostre a codificação para a frase abaixo. Quantos bits você usou?

Pedro Pedreiro Penseiro



8. Considere a árvore abaixo, construída com base no algoritmo de compressão dos códigos de Huffman:



Decodifique a seguinte sequência de bits:

10011011110001011011101110001000001001011011110001011011100011000110101
B O M B A A T O M I C
00101101111000110111000111010110011011110001011011101110001000001

BOMBA ATOMICA