

## Algoritmo CYK (de Cocke, Younger e Kasami)

A bar chart with five bars of decreasing height. Below the bars are labels:  $a$ ,  $b$ ,  $a$ ,  $a$ ,  $b$ .

$A$	$S$	$A$	$A$	$S$
$a$	$b$	$a$	$a$	$b$

Um não terminal  $N$  é colocado na matriz em  $M[i, j]$  a custa de dois outros não terminais  $P$  e  $Q$  se

- $N \rightarrow P Q$  é uma produção.
- $P$  está a esquerda de  $Q$  e ambos estão "por baixo" e a direita de  $N$ .

Dada uma linha  $i$ , o processo vare a linha toda. Repare que este processo pode não poder preencher algumas células, estas ficando assim vazias.

Se repetimos este processo de baixo para cima até chegar ao topo da matriz então podemos determinar se a palavra analisada pertence a linguagem. Para tal basta ver se  $S$  está na célula  $M[1, 1]$ .

Vejamos o processo linha a linha:

	$A, S$	$A$	$S$	$A, S$	
	$A$	$S$	$A$	$A$	$S$
	$a$	$b$	$a$	$a$	$b$

$M[4, 4] = \{A, S\}$  porque  $M[5, 4] = A$ ,  $M[5, 5] = S$ ,  $A \rightarrow AS$  e  $S \rightarrow AS$  são produções.

$M[4, 3] = S$  porque temos  $M[5, 3] = A$  e  $M[5, 4] = A$  e que  $S \rightarrow AA$  é produção.

$M[4, 2] = A$  porque  $M[5, 2] = A$ ,  $M[5, 3] = A$ ,  $A \rightarrow AA$  é produção.

$M[4, 1] = \{A, S\}$  porque  $M[5, 1] = A$ ,  $M[5, 2] = S$ ,  $A \rightarrow AS$  e  $S \rightarrow AS$  são produções.

As linhas seguintes preenchem-se de forma similar.

	$A, S$	$A, S$	$A$		
	$A, S$	$A$	$S$	$A, S$	
	$A$	$S$	$A$	$A$	$S$
	$a$	$b$	$a$	$a$	$b$

	$A, S$	$A, S$			
	$A, S$	$A, S$	$A$		
	$A, S$	$A$	$S$	$A, S$	
	$A$	$S$	$A$	$A$	$S$
	$a$	$b$	$a$	$a$	$b$

	$A, S$				
	$A, S$	$A, S$			
	$A, S$	$A, S$	$A$		
	$A, S$	$A$	$S$	$A, S$	
	$A$	$S$	$A$	$A$	$S$
	$a$	$b$	$a$	$a$	$b$

Neste ponto já podemos concluir:  $abaab$  pertence a linguagem gerada por  $G$ .

Este exemplo é simples porque todas as células da matriz foram preenchidas. Isto nem sempre acontece, como o podemos ver no exemplo seguinte.

Seja  $G' = \{\{S, A\}, \{a, b\}, P, S\}$  com  $P = \{S \rightarrow A B, S \rightarrow b, A \rightarrow a, B \rightarrow S A\}$

	S				
		B			
		S			
	A		B		
	A	S	A	A	
a	a	b	a	a	

De reparar que neste exemplo não se consegue preencher todas as células da matriz. No entanto as células preenchidas levam mesmo assim a que se coloque o símbolo inicial no topo da matriz. A palavra é assim aceite. Na linha 4 só se consegue colocar o  $B$  graças a regra  $B \rightarrow S A$ . De seguida só se consegue colocar o  $S$  na linha 3 porque se tem a regra  $S \rightarrow A B$ . Neste caso o  $A$  provem de  $M[5, 2]$  e  $B$  de  $M[4, 3]$ . O caso realçado na matriz corresponde ao caso da aplicação da regra  $B \rightarrow S A$  com  $A$  retirado de  $M[5, 5]$ . Finalmente consegue-se colocar  $S$  em  $M[1, 1]$  graças ao  $A$  de  $M[5, 1]$  e o  $B$  de  $M[2, 2]$ .

## Input

Na primeira linha é introduzida a palavra por reconhecer. Esta palavra só é constituída por caracteres do alfabeto  $\Sigma = \{a, \dots, z\}$  e tem por comprimento máximo 50 caracteres. Assume-se que o conjunto de não-terminais é  $N = \{A, B, \dots, S, \dots, Z\}$  em que se destaca o símbolo  $S$  que assumiremos como sendo sempre o símbolo inicial.

A segunda linha apresenta o número  $m$  de regras de produções da gramática considerada. As restantes  $m$  linhas introduzem as  $m$  regras de produção. Cada uma destas produções tem o formato seguinte  $N \rightarrow a_1 a_2 a_3 \dots a_n$  com  $N$  não terminal e  $a_i \in (N \cup \Sigma)$  e cada  $a_i$  é separado do  $a_j$  seguinte por um espaço único.

## Output

O output é constituído por uma linha contendo:

- A palavra "YES" se a palavra é gerada pela gramática.
- A palavra "NO" se a palavra não é gerada (reconhecida) pela gramática

Seguida da matriz triangular inferior, usada para reconhecer (ou não) a palavra.

### NOTAS:

- cada célula da matriz é separada da célula adjacente à direita por 2 tabs (`\t\t`).
- Sempre que uma célula contém mais do que um não terminal, estes são separados por um só espaço

### Sample Input 1

```
abbabba
7
S -> S F
S -> a
A -> C C
A -> S S
A -> C S
C -> b
F -> A S
```

### Sample Output 1

```
YES
S
  F
    A
      S
        F
          A
            S
              C
                C
                  S
                    C
                      C
                        S
                          a
                            b
                              b
                                a
                                  b
                                    b
                                      a
```

### Sample Input 2

```
aaabbabaaaabba
8
S -> S F
S -> a
A -> C G
A -> S S
A -> C S
C -> b
F -> A S
G -> C A
```

### Sample Output 2

```
NO

F
  A
    S
      F
        A
          G
            F
              G
                A
                  F
                    G
                      A
                        S
                          S
                            F
                              G
                                F
                                  F
                                    A
                                      G
                                        A
                                          A
                                            A
                                              S
                                                C
                                                  C
                                                    S
                                                      C
                                                        S
                                                          C
                                                            C
                                                              S
                                                                a
                                                                  a
                                                                    a
                                                                      b
                                                                        b
                                                                          a
                                                                            b
                                                                              a
                                                                                a
                                                                                  a
                                                                                    a
                                                                                      a
                                                                                        a
                                                                                          a
                                                                                              a
                                                                                                a
                                                                                                  b
                                                                                                      b
                                                                                                          a
```