

# Problema I

# foraminis punchers

Arquivo fonte: foraminis.{ c | cpp | java | py }
Autor: Prof. Dr. Alex Marino (Fatec Ourinhos)

A guerra entre Sneakyland e Esbórnia persiste há muito tempo. As hostilidades entre os países escalaram a ponto de tornarem-se brutais. O exército de Sneakyland desenvolveu um sistema de mísseis teleguiados que têm causado destruição em alvos estratégicos esbornianos. O sistema, chamado *Foraminis Punchers*, usa o sistema *Phanthon Signum Scrambler* para manter suas trajetórias invisíveis aos radares esbornianos.

A BOSSAD (serviço secreto esborniano), desesperada com a quase certa possibilidade de rendição, deslocou nosso amigo Joãozinho de sua ocupação no desenvolvimento do canhão de laser teletransportador (graças a ele, muitas vidas esbornianas foram salvas) e o incumbiu da missão de criar um sistema anti *Foraminis Punchers*. Como bem conhecemos nosso amigo, missão dada é missão cumprida.

Nosso intrépido amigo descobriu, de maneira desconhecida (Joãozinho tornou-se um excelente espião), que a sequência de sinais digitais transmitidos pelo sistema *Foraminis Punchers* possui uma característica que possibilita identificar sua posição no espaço aéreo esborniano. Essa característica reside na faixa de frequência de ondas curtas, e é possível identificá-la se a energia na faixa baixa de frequência superar 50% da energia do sinal original.

Joãozinho e sua equipe de físicos desenvolveram um dispositivo de filtro de sinal que discretiza o sinal original, com cada sinal de entrada possuindo comprimento/suporte igual a oito, ou seja, suporte M=8. Para identificar o míssil, ainda é preciso criar um filtro passa alta que segregue os componentes de baixa frequência e, por fim, medir sua energia. Para elaborar o filtro passa alta, Joãozinho optou por utilizar o algoritmo de Mallat e usará como convolucionador o par de filtros de Haar.

A explicação de Joãozinho para a solução do problema segue-se: Dado um sinal S[.] = {  $S_0, S_1, S_2, S_3, S_4, S_5, S_6, S_7$ } e o par de filtros Haar, nos quais H={ $h_0, h_1$ } e G={ $g_0, g_1$ }, tem-se :

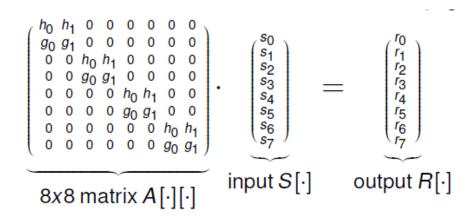


Figura I.1:  $M[.][.] \times S[.] = R[.]$ 

O par de filtros de Haar é composto por: H={  $\frac{1}{\sqrt{2}}$ ,  $\frac{1}{\sqrt{2}}$ } e G={  $\frac{1}{\sqrt{2}}$ ,  $-\frac{1}{\sqrt{2}}$ }.

De tal modo que o sinal resultante R[.]={  $R_0, R_1, R_2, R_3, R_4, R_5, R_6, R_7$ } contém os componentes de alta



(HF) e baixa (LF) frequência conforme explícitos na equação abaixo:

$$i \in HF \quad \forall i\%2 == 1$$
  

$$i \in LF \quad \forall i\%2 == 0$$
(1)

Um vez determinados os sinais HF e LF, podemos facilmente identificar a fragilidade do sistema *foraminis punchers* pela equação abaixo:

$$\begin{cases} \text{identificado}, & \text{se } \frac{E(LF[.])}{E(S[.])} > 0.5, \\ \text{n\~{a}o identificado}, & \text{caso contr\'{a}r\'{i}o}. \end{cases} \tag{2}$$

Por fim, o cálculo da energia de um sinal digital dá-se por:

$$E(S[.]) = \sum_{i=0}^{M-1} S_i^2$$
 (3)

Em que M é o número de componentes do sinal de interesse ou também chamado de suporte do sinal.

Sabemos que Joãozinho não é bom programador e por isso ele te convocou para salvar a população esborniana dos ataques perversos dos endiabrados sneakys. Joãozinho e o povo esborniano contam com sua preciosa ajuda. Assim seu trabalho é implementar o algoritmo explicado.

#### **Entrada**

A primeira linha contém o número N de sinais coletados, tal que N <= 500. Seguida pelas N linhas subsequentes, cujos conteúdos representam os 8 componentes de cada sinal de interesse a ser avaliado.

#### Saída

Para cada sinal de interesse avaliado, seu programa deve responder INIMIGO caso identifique a fragilidade do sinal ou hífen "-" caso contrário.

## Exemplo de Entrada 1

## Exemplo de Saída 1

| 4                       | _       |
|-------------------------|---------|
| 7 -5 4 -3 7 -1 -4 -10   | INIMIGO |
| -6 8 -10 -8 -7 10 -7 -7 | _       |
| 8 -4 1 -10 6 0 2 -4     | _       |
| 1 -8 -5 5 -3 5 6 -7     |         |