Table of Contents

..... 1
Entrada de valores 1
Declaracao das variaveis 1
Logica principal 1


```
% Filosofos Famintos
% Alunos :Danilo Platiny

% Prioridades
% - Sentido horario
% - Quem terminou de comer pensa
% - Quem comeu mais nao come
% - Sempre o maior número de filosofos possiveis irá comer na rodada
% - Quando nao tiver mais comida os filosofos irao esperar o utlimo
 terminar de comer
```

Entrada de valores

```
tempo = [5 2 1 3 5]; % tempo para comer

quantidadeFilosofos = 5; % Numero de Filosofos
```

Declaracao das variaveis

```
%Cria Hashis. Quando 1 os hashis estao sobre a mesa. Quando 0 os
 hashis
%estao sendo utilizados.
hashis = ones(1,quantidadeFilosofos);

% Cria uma matriz de situacao para cada filosofo a dimensao da matriz
 é
% superestimada para nao ocorrer erros.
% 3 = Estado Para Logica interna  2 = Come , 1 = Pensa, 0 = Espera
matrizResultado = 3*ones(quantidadeFilosofos,sum(tempo)) ;

%Cria uma matriz eficiencia com a quantidade de vezes que o filosofo
 pensou
%comeu e esperou
matrizEficiencia = zeros(quantidadeFilosofos,3);

%Variaveis Auxiliares
i = 1; %Controle do vetor tempo
tempoAtual = 1; %Controle das colunas
```

Logica principal

```
%Laco principal. Enquanto tiver comida (vetor de tempo) os filosofos
```

```
%continuam na mesa

while (i <= length(tempo))


%Laco para cada filosofo decidir se irá comer
    for a = 1:1:quantidadeFilosofos

     %Condicoes que só ocorrem depois da primeira rodada
        if(tempoAtual~= 1)
       %Se o filosofo tem que comer essa rodada pula para proximo
 filosofo
            if(matrizResultado(a,tempoAtual) == 2)
                continue
            end
        %Se o filosofo comeu na ultima rodada ira pensar nessa
            if (matrizResultado(a,tempoAtual -1) == 2)
                [matrizEficiencia, matrizResultado] =
 pensa(matrizResultado, matrizEficiencia, a,tempoAtual); % Atualiza as
 tabelas de resultado
                continue
            end
        %Descobre quem mais comeu ate a rodada atual
           filosofoComilao =
 oQueMaisComeu(matrizEficiencia,quantidadeFilosofos);

        end

      % Filosofo olha se os hashis estao disponiveis
            situacaoHashis = olharHashis(hashis, a);

        %Filosofo pergunta se ele pode comer pelas prioridades
            oqueFazer = conferePrioridade(situacaoHashis,
 filosofoComilao,a);

        % Se ele poder comer
            if(strcmp(oqueFazer,'podeComer'))
                hashis = atualizaHashis(hashis,a,'pega'); % Pega os
 hashis
                [matrizEficiencia, matrizResultado]  =
 come(matrizResultado, matrizEficiencia, tempo, a, i,tempoAtual); %
 Atualiza as tabelas de resultado
                i = i + 1; % Anda no vetor de tempo

            end
   end

  % Depois de definir quem irá comer irá determinar quem irá pensar e
 quem
  % irá esperar
   for a = 1:1:quantidadeFilosofos

      % Se no for anterior foi determinado para o filosofo pensar ou
 comer
```

```
        % ele nao irá passar por esse loop.
        if(matrizResultado(a,tempoAtual) == 3)

        % Filosofo olha se os hashis estao disponiveis
             situacaoHashis = olharHashis(hashis, a);

         %Filoso pergunta se ele pode comer pelas prioridades
             oqueFazer = conferePrioridade(situacaoHashis,
 filosofoComilao,a);

         % Se ele tiver que esperar
             if(strcmp(oqueFazer,'podeEsperar'))
                 [matrizEficiencia, matrizResultado] =
 espera(matrizResultado, matrizEficiencia, a,tempoAtual); % Atualiza
 as tabelas de resultado

          % Se ele tiver que pensar
             elseif(strcmp(oqueFazer,'podePensar'))
                 [matrizEficiencia, matrizResultado] =
 pensa(matrizResultado, matrizEficiencia, a,tempoAtual); % Atualiza as
 tabelas de resultado
             end

       end
    end


    %Avanca uma unidade de tempo
    if(tempoAtual <  size(matrizResultado,2) ) % regra para nao
 permetir que estoure o tamnanho da matriz
         tempoAtual = tempoAtual +1;
    end
    %Confere se algum filosofo deve soltar os hashis para proxima
 rodada
    for b = 1:1:quantidadeFilosofos
        if(matrizResultado(b,tempoAtual) ~= 2)
            if(matrizResultado(b,tempoAtual -1) == 2)
                hashis = atualizaHashis(hashis,b,'solta');
            end
        end
    end

end

%Define em que tempo o ultimo terá terminado de comer
aux = tempoAtual;
for t= 1:tempoAtual:size(matrizResultado,2)
    for b = 1:1:quantidadeFilosofos
        if(matrizResultado(b,aux)== 2)
            aux = aux +1;
            continue
        end
    end
end
```

```
%Ordena que os filosofos esperem até o ultimo terminar de comer
while(tempoAtual ~= aux+1)
    for b = 1:1:quantidadeFilosofos
        if(matrizResultado(b,tempoAtual) ~= 2)

            if(matrizResultado(b,tempoAtual -1) == 2) % Pensa se
 acabou de comer
                [matrizEficiencia, matrizResultado] =
 pensa(matrizResultado, matrizEficiencia, b,tempoAtual);% Atualiza as
 tabelas de resultado
                continue;
            end

            [matrizEficiencia, matrizResultado] =
 espera(matrizResultado, matrizEficiencia, b,tempoAtual); % Atualiza
 as tabelas de resultado
        end
    end
    tempoAtual = tempoAtual +1;
end

% Elimna a parte da matriz que nao apresenta resultado
matrizResultadoFinal = matrizResultado(:,1:10)
matrizEficiencia
```

```
Undefined function or variable 'filosofoComilao'.

Error in FilosofosFamintos (line 68)
            oqueFazer = conferePrioridade(situacaoHashis,
 filosofoComilao,a);
```

Published with MATLAB® R2015a