Enunciado

Se pretende, teniendo un almacén 2D (sin pérdida de generalidad), realizar un programa que simule la atención a un determinado pedido de material. Los materiales se encuentran etiquetados con números naturales y cada uno lleva un determinado tiempo almacenado en el almacén.

El almacén podría “verse”, por ejemplo, como un patio de contenedores de una altura.

El pedido contiene los identificadores de los productos a retirar. El programa debe decirnos si se consigue satisfacer el pedido y mostrarnos la situación del almacén a cada paso de la atención del pedido.

En cada unidad de tiempo, además debe actualizarse la matriz de tiempos de los materiales.

En esta primera versión, se usan dos estrategias de atención a los pedidos:

- FIFO: se extrae de almacén el producto solicitado que lleve más tiempo
- Aleatoria: se extrae del almacén cualquier producto pedido de forma aleatoria

Implementación

`SimulaAlmacen.m`: script para inicializar el almacén en cuanto a productos, tiempos almacenados y pedido. Todo se inicializa de forma aleatoria.

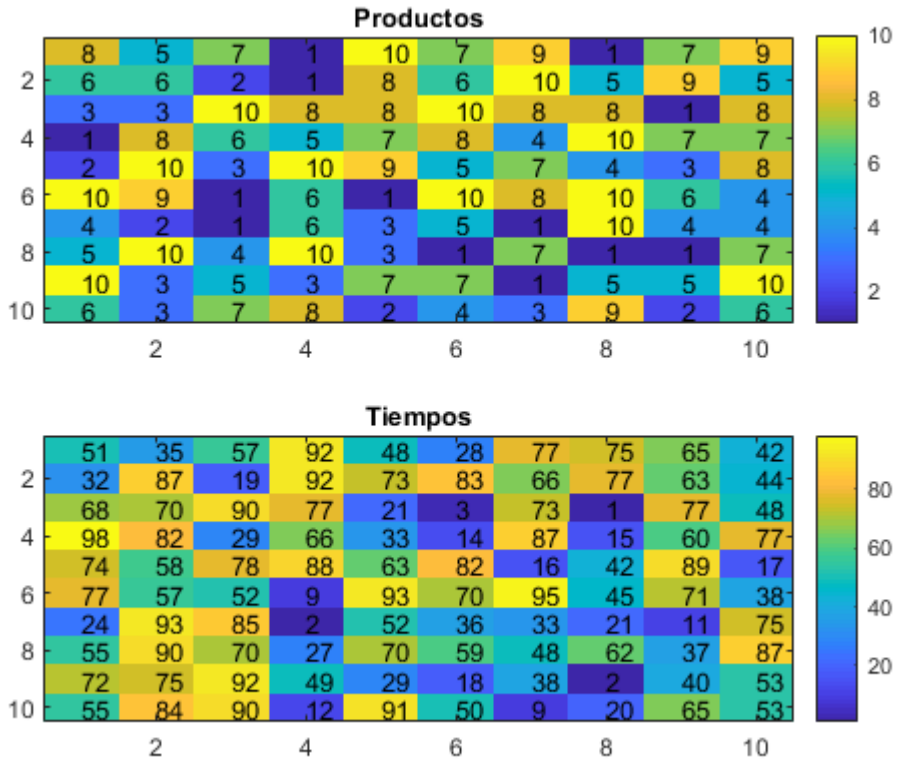
`Almacen.m`: subprograma para atender el pedido.

`DibujaAlmacen.m`: subprograma para dibujar los productos que están en el almacén y sus tiempos de almacenamiento.

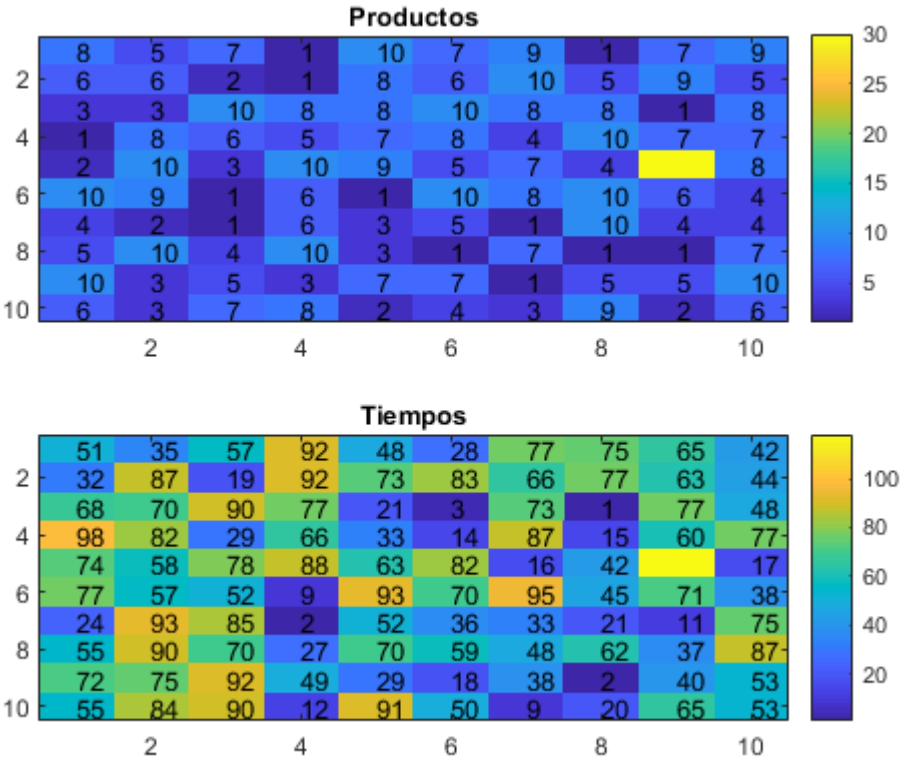
```
% SCRIPT para simular un almacen
% Se tiene una matriz MxN que representa un almacén, en la que cada celda contiene un
número de producto, y el tiempo que lleva almacenado.
% Se pide, para un pedido de material que se solicite (representado por una tabla de
productos),
% determinar el éxito al intentar satisfacerlo, las coordenadas de las localizaciones de
los productos salientes en el almacén,
% y el tiempo tardado al atenderlo:
% a)    si generamos aleatoriamente localizaciones de la matriz que coincidan con cada
producto que hay que extraer del almacén.
% b)    si decidimos que debe salir en cada caso, la unidad que lleve más tiempo en el
almacén.
%
% Ejemplo:
% -     Almacén 4x3: cada celda contiene stock (Producto, tiempo)
% (3,3) (3,2) (4,10)
% (2,5) (2,5) (4, 15)
% (1,5) (1,6) (1,7)
% (3,1) (3,1) (4,20)
%
% -     Pedido: 1 1 3
% a)    coordenadas (3,1) (3,3) (1,2). Éxito.
% b)    coordenadas (3,3) (3,2) (1,1). Éxito.


clear all;
close all;

% longitud del pedido
P = 5;
% dimensión del almacen
M = 10;
N = 10;
% número de productos
NP = 10;

producto = floor(NP*rand(M,N))+1;
tiempo   = floor(100*rand(M,N))+1;
pedido   = floor(NP*rand(P,1))+1;


DibujaAlmacen(producto,tiempo,100,100);

% figure,image(pedido),title('Pedido');

[exito,locs,time] = Almacen(producto,tiempo,pedido,'fifo');


% Escritura en pantalla de las localizaciones de los productos
% del pedido en el almacen
fig1 = figure('pos', [50 350 500 300]);
set(gca,'Visible','off');
z0 = 0.00;
y0 = 0.85;
dz = 0.15;
```

```
dy = 0.06;
text(z0+0.40,y0,'Localizaciones en almacen')
z = z0+dz;
y = y0-3*dy/2;
colheads = ['X                  ';'Y                  ';'Producto           '];

for i =1:3
   text(z,y,colheads(i,:))
  z = z + dz;
end
s = sprintf('Tiempo = %d',time);
xlabel(s);

for i=1:P
  y = y0-(i+1.5)*dy;
  z = z0 + dz;
  h = text(z,y,[sprintf('%2d          ',locs(i,1)) sprintf('\t\t\t%2d
',locs(i,2)) sprintf('\t--->          %2d',pedido(i)) ]);
  z = z + dz;
end
```

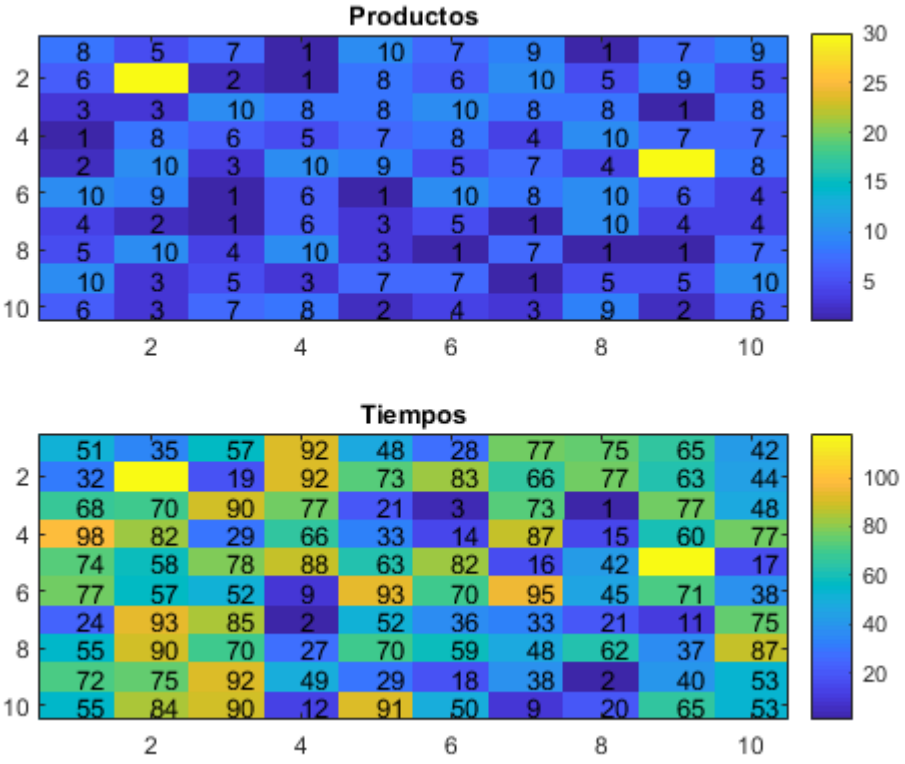
Pedido:
3,6,3,7,1



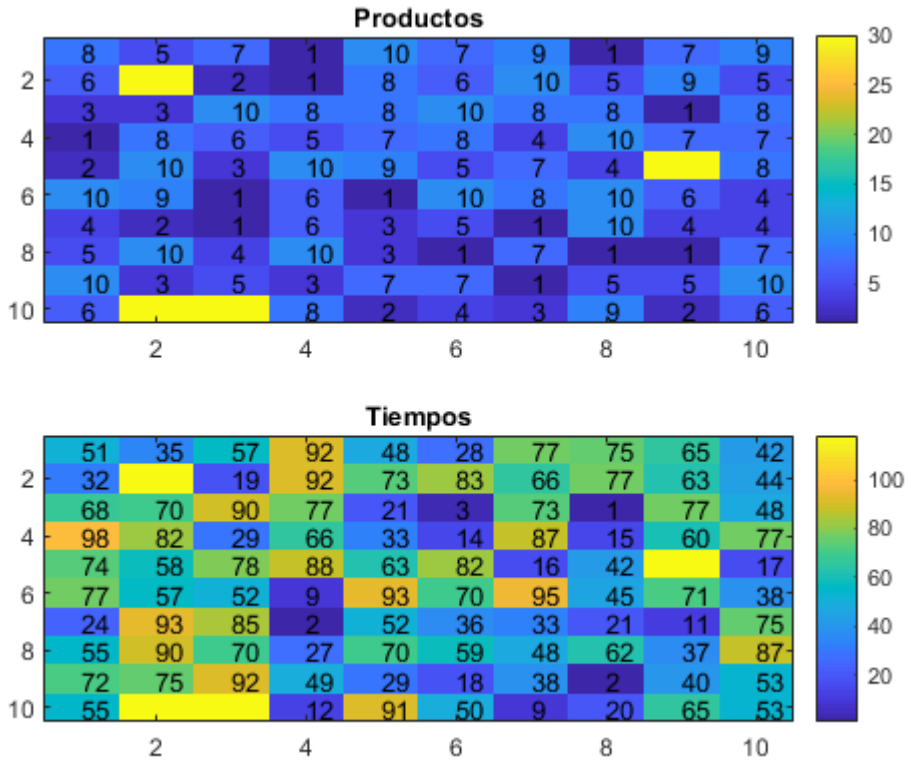
Paso
1


Productos
8 5 7 1 10 7 9 1 7 9
6 6 2 1 8 6 10 5 9 5
3 3 10 8 8 10 8 8 1 8
1 8 6 5 7 8 4 10 7 7
2 10 3 10 9 5 7 4 8
10 9 1 6 1 10 8 10 6 4
4 2 1 6 3 5 1 10 4 4
5 10 4 10 3 1 7 1 1 7
10 3 5 3 7 7 1 5 5 10
6 3 7 8 2 4 3 9 2 6
2 4 6 8 10
30 25 20 15 10 5
Tiempos
51 35 57 92 48 28 77 75 65 42
32 87 19 92 73 83 66 77 63 44
68 70 90 77 21 3 73 1 77 48
98 82 29 66 33 14 87 15 60 77
74 58 78 88 63 82 16 42 17
77 57 52 9 93 70 95 45 71 38
24 93 85 2 52 36 33 21 11 75
55 90 70 27 70 59 48 62 37 87
72 75 92 49 29 18 38 2 40 53
55 84 90 12 91 50 9 20 65 53
2 4 6 8 10
100 80 60 40 20

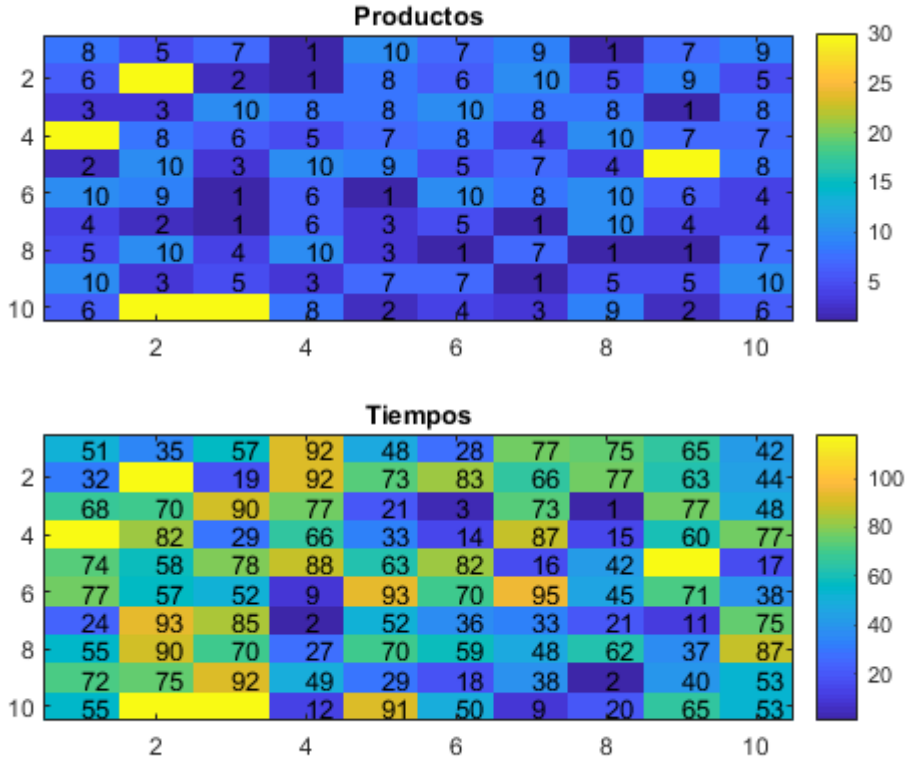
Paso
2


Productos
8 5 7 1 10 7 9 1 7 9
6 2 1 8 6 10 5 9 5
3 3 10 8 8 10 8 8 1 8
1 8 6 5 7 8 4 10 7 7
2 10 3 10 9 5 7 4 8
10 9 1 6 1 10 8 10 6 4
4 2 1 6 3 5 1 10 4 4
5 10 4 10 3 1 7 1 1 7
10 3 5 3 7 7 1 5 5 10
6 3 7 8 2 4 3 9 2 6
2 4 6 8 10
30 25 20 15 10 5
Tiempos
51 35 57 92 48 28 77 75 65 42
32 19 92 73 83 66 77 63 44
68 70 90 77 21 3 73 1 77 48
98 82 29 66 33 14 87 15 60 77
74 58 78 88 63 82 16 42 17
77 57 52 9 93 70 95 45 71 38
24 93 85 2 52 36 33 21 11 75
55 90 70 27 70 59 48 62 37 87
72 75 92 49 29 18 38 2 40 53
55 84 90 12 91 50 9 20 65 53
2 4 6 8 10
100 80 60 40 20

Paso 3


Productos
8 5 7 1 10 7 9 1 7 9
6 2 1 8 6 10 5 9 5
3 3 10 8 8 10 8 8 1 8
1 8 6 5 7 8 4 10 7 7
2 10 3 10 9 5 7 4 8
10 9 1 6 1 10 8 10 6 4
4 2 1 6 3 5 1 10 4 4
5 10 4 10 3 1 7 1 1 7
10 3 5 3 7 7 1 5 5 10
6 8 2 4 3 9 2 6
2 4 6 8 10
30 25 20 15 10 5
Tiempos
51 35 57 92 48 28 77 75 65 42
32 19 92 73 83 66 77 63 44
68 70 90 77 21 3 73 1 77 48
98 82 29 66 33 14 87 15 60 77
74 58 78 88 63 82 16 42 17
77 57 52 9 93 70 95 45 71 38
24 93 85 2 52 36 33 21 11 75
55 90 70 27 70 59 48 62 37 87
72 75 92 49 29 18 38 2 40 53
55 12 91 50 9 20 65 53
2 4 6 8 10
100 80 60 40 20

Paso
4


Productos
8 5 7 1 10 7 9 1 7 9
6 2 1 8 6 10 5 9 5
3 3 10 8 8 10 8 8 1 8
8 6 5 7 8 4 10 7 7
2 10 3 10 9 5 7 4 8
10 9 1 6 1 10 8 10 6 4
4 2 1 6 3 5 1 10 4 4
5 10 4 10 3 1 7 1 1 7
10 3 5 3 7 7 1 5 5 10
6 8 2 4 3 9 2 6
2 4 6 8 10
30 25 20 15 10 5
Tiempos
51 35 57 92 48 28 77 75 65 42
32 19 92 73 83 66 77 63 44
68 70 90 77 21 3 73 1 77 48
82 29 66 33 14 87 15 60 77
74 58 78 88 63 82 16 42 17
77 57 52 9 93 70 95 45 71 38
24 93 85 2 52 36 33 21 11 75
55 90 70 27 70 59 48 62 37 87
72 75 92 49 29 18 38 2 40 53
55 12 91 50 9 20 65 53
2 4 6 8 10
100 80 60 40 20

Localizaciones en almacén:

X	Y	Producto
5	9	3
2	2	6
10	2	3
10	3	7
4	1	1

```
function [exito,locs,t] = Almacen(producto,tiempo,pedido,estrategia)
% ALMACEN simula un almacen con dos matrices con los números de producto
% y el tiempo que lleva almacenado. Se pide, para un pedido de material que
% se solicite (representado por una tabla de productos), determinar el
% éxito al intentar satisfacerlo, las coordenadas de las localizaciones de
% los productos salientes en el almacén, y el tiempo tardado al atenderlo:
% a)    si generamos aleatoriamente localizaciones de la matriz que
% coincidan con cada producto que hay que extraer del almacén.
% b)    si decidimos que debe salir en cada caso, la unidad que lleve
% más tiempo en el almacén. FIFO.
%
% Ejemplo:
% -     Almacén 4x3: cada celda contiene (Producto, tiempo)
% (3,3) (3,2) (4,10)
% (2,5) (2,5) (4, 15)
% (1,5) (1,6) (1,7)
% (3,1) (3,1) (4,20)
%
% -     Pedido: 1 1 3
% a)    coordenadas (3,1) (3,3) (1,2). Éxito.
% b)    coordenadas (3,3) (3,2) (1,1). Éxito.
%
% ENTRADAS:
%   producto,tiempo : matrices que representan el almacen
%   pedido          : vector que almacena un pedido de productos
%   estrategia      : 'alea' (aleatoria), 'fifo'
% SALIDAS:
%   exito           : 1 si se satisface el pedido, 0 en caso contrario
%   locs            : posiciones de los productos en el almacen
%   t               : tiempo de servicio del pedido, considerando la
%                     distancia desde el origen (se considera el punto 0,0


[M,N] = size(producto);
P = length(pedido);

% Para dibujar ------------------------------------------------------
iprod = producto;
itiempo = tiempo;
max_iprod = max(max(producto));
max_itiempo = max(max(tiempo));
% -------------------------------------------------------------------

% Recorrido del pedido para ver si se puede satisfacer
exito = 1;
i = 1;
t = 0;
actual_x = 0;
actual_y = 0;

while (exito == 1) && (i <= P)
    [posx,posy] = find(producto==pedido(i));
    l = length(posx);
```

```
    if l == 0
        exito = 0;
    else
        if estrategia == 'alea'
            a = floor(l*rand(1))+1;
            locs(i,:) = [posx(a) posy(a)];
            d = sqrt((posx(a)-actual_x)^2+(posy(a)-actual_y)^2);
            t = t + d;
            actual_x = posx(a);
            actual_y = posy(a);
            producto(posx(a),posy(a)) = -1;
            tiempo(posx(a),posy(a)) = 0;
            % Para dibujar ------------------------------------------------
            % Al máximo valor le sumamos 20 por visualización
            iprod(posx(a),posy(a)) = max_iprod + 20;
            itiempo(posx(a),posy(a)) = max_itiempo + 20;
            DibujaAlmacen(iprod,itiempo,max_iprod + 20,max_itiempo + 20);
            % ------------------------------------------------------------
            i = i + 1;
        end
        if estrategia == 'fifo'
            maxi = -10000;
            for a = 1:l
                if tiempo(posx(a),posy(a)) > maxi
                    maxi = tiempo(posx(a),posy(a));
                    pos_max = a;
                end
            end
            locs(i,:) = [posx(pos_max) posy(pos_max)];
            d = sqrt((posx(pos_max)-actual_x)^2+(posy(pos_max)-actual_y)^2);
            t = t + d;
            actual_x = posx(pos_max);
            actual_y = posy(pos_max);
            producto(posx(pos_max),posy(pos_max)) = -1;
            tiempo(posx(pos_max),posy(pos_max)) = 0;
             % Para dibujar ------------------------------------------------
            % Al máximo valor le sumamos 20 por visualización
            iprod(posx(pos_max),posy(pos_max)) = max_iprod + 20;
            itiempo(posx(pos_max),posy(pos_max)) = max_itiempo + 20;
            DibujaAlmacen(iprod,itiempo,max_iprod + 20,max_itiempo + 20);
            % ------------------------------------------------------------
            i = i + 1;
        end
        % posibilidad de implementar otras estrategias
    end
end
```

Published with MATLAB® R2018b