L19. Two-Dimensional Arrays

Set-Up
Rows and Columns
Subscripting
Operations
Examples
Simple Set-Up Examples

\[
\begin{bmatrix}
    1 & 2 & 3 \\
    4 & 5 & 6
\end{bmatrix}
\]
Simple Set-Up Examples

\[
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{bmatrix}
\]
Simple Set-Up Examples

```matlab
>> A = floor(100*rand(5,5))
A =
   95    76    61    40     5
   23    45    79    93    35
   60     1    92    91    81
   48    82    73    41     0
   89    44    17    89    13
```
Simple Set-Up Examples

```
>> A = [zeros(3,2) [1;2;3]]
A =

0     0     1
0     0     2
0     0     3
```
Simple Set-Up Examples

```matlab
>> A = [zeros(3,2) ; [1 2] ]
A =
    0     0
    0     0
    0     0
    0     0
    1     2
```
## Rows and Columns

A is a 3-by-4 array: 3 rows 4 columns.
Subscripting

<table>
<thead>
<tr>
<th>A:</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>17</td>
<td>49</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>18</td>
<td>82</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>83</td>
<td>53</td>
<td>12</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

Individual entries: $A(3, 2)$
Subscripting

\[
A = \begin{bmatrix}
12 & 17 & 49 & 61 \\
38 & 18 & 82 & 77 \\
83 & 53 & 12 & 10
\end{bmatrix}
\]

An Entire Row: \(A(2,:)\)
Scaling a Row

$$A(2,:) = 10\times A(2,:).$$

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>12</td>
<td>17</td>
<td>49</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>83</td>
<td>53</td>
<td>12</td>
<td>10</td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>12</td>
<td>17</td>
<td>49</td>
<td>61</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>83</td>
<td>53</td>
<td>12</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>
**Subscripting**

An entire column: \( A(:,3) \)

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>17</td>
<td>49</td>
<td>61</td>
</tr>
<tr>
<td>38</td>
<td>18</td>
<td>82</td>
<td>77</td>
</tr>
<tr>
<td>83</td>
<td>53</td>
<td>12</td>
<td>10</td>
</tr>
</tbody>
</table>
Incrementing the Values in a Column

A:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>17</td>
<td>49</td>
<td>61</td>
</tr>
<tr>
<td>38</td>
<td>18</td>
<td>82</td>
<td>77</td>
</tr>
<tr>
<td>83</td>
<td>53</td>
<td>12</td>
<td>10</td>
</tr>
</tbody>
</table>

A:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>17</td>
<td>50</td>
<td>61</td>
</tr>
<tr>
<td>38</td>
<td>18</td>
<td>83</td>
<td>77</td>
</tr>
<tr>
<td>83</td>
<td>53</td>
<td>13</td>
<td>10</td>
</tr>
</tbody>
</table>

A(:,3) = A(:,3) + 1

Before

After
Subscripting

A General Subarray: \( A(2:3,3:4) \)
Zeroing a Subarray

A:

Before

A(2:3,3:4) = zeros(2,2)

After

A:

\[
\begin{bmatrix}
12 & 17 & 49 & 61 \\
38 & 18 & 82 & 77 \\
83 & 53 & 12 & 10 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
12 & 17 & 49 & 61 \\
38 & 18 & 0 & 0 \\
83 & 53 & 0 & 0 \\
\end{bmatrix}
\]
### Classical Double Loop Set-Up

\[
\begin{array}{cccc}
11 & 21 & 31 & 41 \\
12 & 22 & 32 & 42 \\
13 & 23 & 33 & 43 \\
\end{array}
\]

```matlab
for i=1:3
    for j=1:4
        A(i,j) = 10*j + i;
    end
end
end
```
Set-Up By Row

<table>
<thead>
<tr>
<th></th>
<th>11</th>
<th>21</th>
<th>31</th>
<th>41</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>22</td>
<td>32</td>
<td>42</td>
</tr>
<tr>
<td>2</td>
<td>13</td>
<td>23</td>
<td>33</td>
<td>43</td>
</tr>
</tbody>
</table>

\[
A = []; \\
for i=1:3 \\
    v = [10 20 30 40] + i; \\
    A = [A ; v] \\
end
\]
Set-Up By Column

A:

\[
\begin{array}{cccc}
11 & 21 & 31 & 41 \\
12 & 22 & 32 & 42 \\
13 & 23 & 33 & 43 \\
\end{array}
\]

A = [];  
for j=1:4  
v = 10*j + [1;2;3];  
A = [A v]  
end
Question Time

\[ A = \begin{bmatrix} 1 & 2 & 3; & 4 & 5 & 6 \end{bmatrix}; \]
\[ C = A(:,2); \]

What the value of \( A(2,2) \)?

A. 4  B. 5  C. 6
Question Time

A = [ 1 2 3; 4 5 6];
A = A(1:2,2:3)

What the value of A(2,2)?

A. 4  B. 5  C. 6
### Largest Value

**A:**

<table>
<thead>
<tr>
<th></th>
<th>12</th>
<th>17</th>
<th>49</th>
<th>61</th>
</tr>
</thead>
<tbody>
<tr>
<td>38</td>
<td>18</td>
<td>82</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>83</td>
<td>53</td>
<td>12</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

**m:**

| 83 | 53 | 82 | 77 |

**M:**

| 83 |

\[ m = \max(A) ; \quad M = \max(m) \]
Functions and 2D Arrays

function alpha = Ave(A)

% A is a 2D array.
% alpha is the average of its values.

10 20 30
40 50 60

-> (10+20+30+40+50+60)/6
function alpha = Ave(A)
    [m,n] = size(A);
    s = sum(A); % Add up all the numbers in the array. Store in s.
    alpha = s/(m*n);

size(A) returns #rows and # columns
function alpha = Ave(A)
    [m,n] = size(A);
    s = 0;
    for i=1:m
        sRow = the sum of the values in A(i,:)
        s = s + sRow
    end
    alpha = s/(m*n);
sRow = the sum of the values in A(i,:)

sRow = 0;
for j=1:n
    sRow = sRow + A(i,j);
end
function alpha = Ave(A)
 [m,n] = size(A);
 s = 0;
 for i=1:m
     sRow = 0;
     for j=1:n
         sRow = sRow + A(i,j);
     end
     s = s + sRow
 end
 alpha = s/(m*n);
Now Some More Involved Examples
Random Web

N web pages

N-by-N Link Array A.

A(i,j) is 1 if there is a link on webpage j to webpage i

Generate a random link array and display the connectivity.
Random Link Idea

\[ A(i, j) = 1 \text{ with probability } \frac{1}{1 + |i - j|} \]

More likely to be a link if \( i \) is close to \( j \).
function A = RandomLinks(n)
A = zeros(n,n);
for i=1:n
    for j=1:n
        r = rand;
        if i~=j && r<= 1/(1 + abs(i-j));
            A(i,j) = 1;
        end
    end
end
\( N = 20 \)

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
100 Web pages. Now display the links....
Line black as it leaves page \( j \), red when it arrives at page \( i \).