

- Previous Lecture:
 - Structure, structure array
 - Working with data files
- Today's Lecture:
 - Another data file example; built-in `sort` function
 - Review matrix, cell array, structure array
- Announcement:
 - Discussion in classrooms this week
 - Project 5 due Thurs 11pm
 - Prelim 3 on Tues, Nov 16th, at 7:30pm. Email Randy Hess **now** if you have a prelim conflict

2010

A detailed sort-a-file example

Suppose each line in the file `statePop.txt` is structured as follows:

Cols 1-14: State name
Cols 16-24: Population (millions)

The states appear in alphabetical order.

Lecture 21

3

A detailed sort-a-file example

Create a new file

`statePopSm2Lg.txt`

that is structured the same as `statePop.txt` except that the states are ordered from smallest to largest according to population.

Alabama	4557808
Alaska	663661
Arizona	5939292
Arkansas	2779154
California	36132147
Colorado	4665177
:	:
:	:

- Need the pop as *numbers* for sorting.
- Can't just sort the pop—have to maintain association with the state names.

Lecture 21

5

```
function CA = file2cellArray(fname)
% fname is a string that names a .txt file
% in the current directory.
% CA is a cell array with CA{k} being the
% k-th line in the file.
```

```
fid= fopen([fname '.txt'], 'r');
k= 0;
while ~feof(fid)
    k= k+1;
    CA{k}= fgetl(fid);
end
fclose(fid);
```

Lecture 21

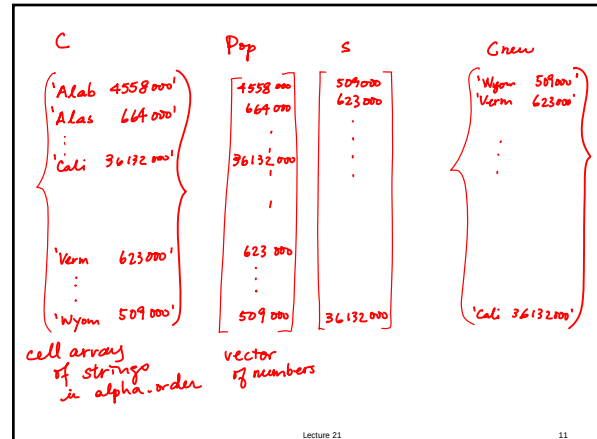
7

First, get the populations into an array

```
C = file2cellArray('StatePop');
n = length(C);
pop = zeros(n,1);
for i=1:n
    S = C{i};
    pop(i) = str2double(S(16:24));
end
```

Lecture 21

8



Lecture 21

11

Built-In function **sort**Syntax: `[y,idx] = sort(x)`x:

10	20	5	90	15
----	----	---	----	----

y:

5	10	15	20	90
---	----	----	----	----

idx:

3	1	5	2	4
---	---	---	---	---

`y(1) = x(3) = x(idx(1))`

Lecture 21

12

Built-In function **sort**Syntax: `[y,idx] = sort(x)`x:

10	20	5	90	15
----	----	---	----	----

y:

5	10	15	20	90
---	----	----	----	----

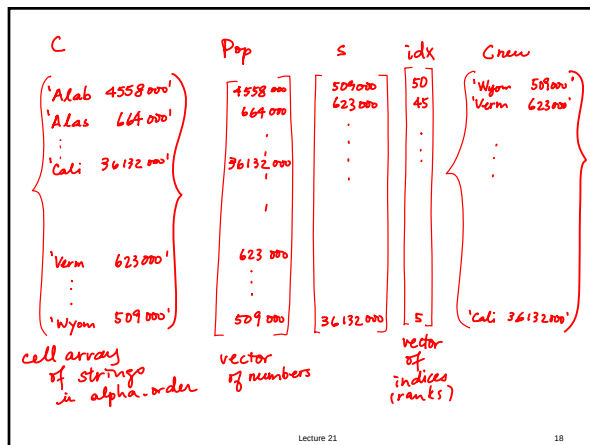
idx:

3	1	5	2	4
---	---	---	---	---

`y(k) = x(idx(k))`

Lecture 21

17



Lecture 21

18

Sort from little to big

```
% C is cell array read from statePop.txt
% pop is vector of state pop (numbers)
[s,idx] = sort(pop);
Cnew = cell(n,1);
for i=1:length(C)
    ithSmallest = idx(i);
    Cnew{i} = C{ithSmallest};
end

cellArray2file(Cnew,'statePopSm2Lg')
```

Lecture 21

19

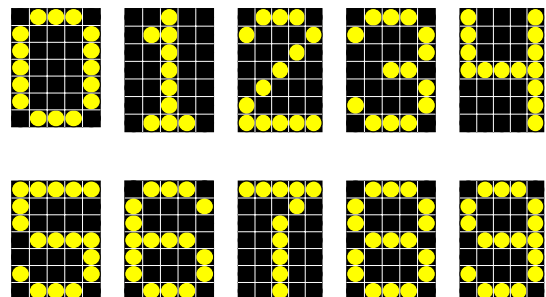
Application: digital displays



Lecture 21

21

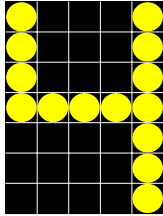
7-by-5 "dot matrices"



Lecture 21

22

A “bit map” for each digit



A “light” is either on or off.

A 7-by-5 matrix of zeros and ones can “tell the whole story.”

Lecture 21

23

Computing with these bitmaps

- What is a good scheme for storing the 10 bitmaps?
- How to draw one digit?
- How to display a number?
- Other interesting questions:
 - How to draw a mirror image of a digit?
 - Which “light bulb” switches on most often?

Lecture 21

24

Design decisions...

How do we package a particular digit?

numerical array or character array

How do we package the collection of digits?

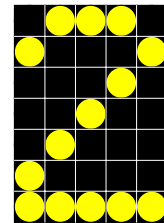
cell array or structure array

Lecture 21

25

Can use a numerical array for each digit

```
[ 0 1 1 1 0;...
  1 0 0 0 1;...
  0 0 0 1 0;...
  0 0 1 0 0;...
  0 1 0 0 0;...
  1 0 0 0 0;...
  1 1 1 1 1];
```

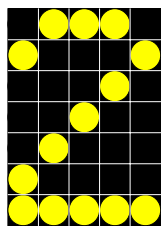


Lecture 21

26

Can use a character array for each digit

```
[ '01110';...
  '10001';...
  '00010';...
  '00100';...
  '01000';...
  '10000';...
  '11111'];
```

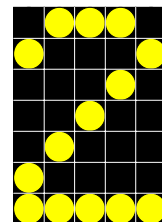


Lecture 21

27

Can use a cell array to keep the 10 bitmaps

```
M = [ 0 1 1 1 0;...
      1 0 0 0 1;...
      0 0 0 1 0;...
      0 0 1 0 0;...
      0 1 0 0 0;...
      1 0 0 0 0;...
      1 1 1 1 1];
```



```
D{2} = M;
```

Each cell of cell array **D** is a numerical matrix.

Lecture 21

28

With $D\{1\}, \dots, D\{10\}$ set up as a **cell array** of **numerical matrices**, can do computation as follows:

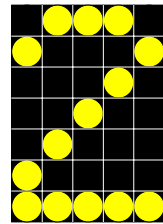
```
% given 1<=k<=10
M = D{k};
if M(4,3)==1
    disp('Middle light is on')
end
```

Lecture 21

29

Still using a **cell array** to keep the 10 bitmaps

```
M = ['01110';...
     '10001';...
     '00010';...
     '00100';...
     '01000';...
     '10000';...
     '11111'];
```



```
D{2} = M;
```

Each cell of cell array **D** is a **matrix of characters**.

Lecture 21

30

With $D\{1\}, \dots, D\{10\}$ set up as a **cell array** of **character matrices**, can do computation as follows:

```
% given 1<=k<=10
M = D{k};
if strcmp(M(4,3), '1')
    disp('Middle light is on')
end
```

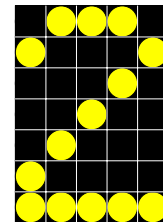
```
['01110';...
 '10001';...
 '00010';...
 '00100';...
 '01000';...
 '10000';...
 '11111'];
```

Lecture 21

31

Can use a **structure array** to keep the 10 bitmaps

```
M = [ 0 1 1 1 0;...
      1 0 0 0 1;...
      0 0 0 1 0;...
      0 0 1 0 0;...
      0 1 0 0 0;...
      1 0 0 0 0;...
      1 1 1 1 1];
```



```
D(2) = struct('map', M);
```

Each component of array **D** is a **structure**, and the sole field in the structure is a **matrix of numbers**.

Using a **structure array** to keep the 10 bitmaps...
The k -th component of array **D** encodes digit k .

```
% Example for the digit 2
A = [ 0 1 1 1 0;...
      1 0 0 0 1;...
      0 0 0 1 0;...
      0 0 1 0 0;...
      0 1 0 0 0;...
      1 0 0 0 0;...
      1 1 1 1 1];
D(2)= struct('map', A);
```

Which fragment on the right is correct given $1 \leq k \leq 10$?

```
M = D{k};
if M(4,3)==1
    disp('Middle light on')
end
```

A

```
M = D{k}.map;
if M(4,3)==1
    disp('Middle light on')
end
```

B

```
M = D{k};
if M(4,3)==1
    disp('Middle light on')
end
```

C

```
M = D{k}.map;
if M(4,3)==1
    disp('Middle light on')
end
```

D

With $D(1), \dots, D(10)$ set up as a **structure array** of **numerical matrices**, can do computation as follows:

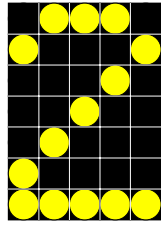
```
% given 1<=k<=10
M = D(k).map;
if M(4,3)==1
    disp('Middle light is on')
end
```

Lecture 21

35

Can use a **structure array** to keep the 10 bitmaps

```
M = [ '01110';...
      '10001';...
      '00010';...
      '00100';...
      '01000';...
      '10000';...
      '11111'];
```



```
D(2) = struct('map', M);
```

Each component of array **D** is a structure, and the sole field in the structure is a **matrix of characters**.

With **D(1),...,D(10)** set up as a **structure array of character matrices**, can do computation as follows:

```
% given 1<=k<=10
M = D(k).map;
if strcmp(M(4,3),'1')
    disp('Middle light is on')
end
```

```
[ '01110';...
  '10001';...
  '00010';...
  '00100';...
  '01000';...
  '10000';...
  '11111'];
```

Lecture 21

37

Choice for storing the bit maps

Cell array better than struct array

No point in having a structure with one field

Numerical array better than char array

Plan on doing numerical computations with the bit maps—char arrays not handy

Lecture 21

38

```
function D = TheDigits()
% D is a 10-by-1 cell array where D{k} is a 7-by-5 matrix
% that encodes digit k. D{10} encodes 0.
```

```
D = cell(10,1);
```

```
D{1} = [0 0 1 0 0;...
        0 1 1 0 0;...
        0 0 1 0 0;...
        0 0 1 0 0;...
        0 0 1 0 0;...
        0 0 1 0 0;...
        0 1 1 0 0];
```

```
D{2} = [0 1 1 1 0;...
        1 0 0 0 1;...
        0 0 0 0 1;...
        0 0 0 1 0;...
        0 0 1 0 0;...
        0 1 0 0 0;...
        1 1 1 1 1];
```

```
:
```

Given this function, can write other functions to draw a single digit, multiple digits, etc.

See also **showDigits.m**

Lecture 21

39

Produce a cell array of “reverse” digits



For every digit (matrix), need to reverse the order of the columns.

Lecture 21

40

```
function B = reverseCol(A)
```

```
% B is a matrix obtained by reversing
% the order of the columns in matrix A
```

```
[nr, nc]= size(A);
B= zeros(nr,nc);
for k= 1:nc
    B(:,k) = A(:,nc-k+1);
end
```

Lecture 21

41

```
function revD = reverseDigits()
% revD is a 10-by-1 cell array.
% revD{k} is the reversed 7-by-5 bitmap
% of digit k. revD{10} encodes 0.
```

```
D= TheDigits();
revD= cell(10,1);
```

```
for k= 1:length(D)
    M= D{k};
    revM= reverseCol(M);
    revD{k}= revM;
end
```

Digital display of a whole number

- Example: **showNumber(2010)**



- Need to convert the number to a vector of digits
 - 2010 → [2 0 1 0]
- Then display the digits in the vector side-by-side

Lecture 21

44

```
function showNumber(n)
% Digital display of integer n, n>0

hold on; axis equal off

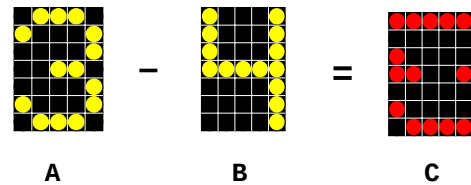
% Convert n to a vector of digits

% Display the digits in v
D = TheDigits(); % D{k} is matrix encoding digit k
for k=1:length(v)
    index= v(k);
    if index==0
        index= 10;
    end
    drawDigit(k,1,1,D{index})
end
```

Lecture 21

45

How to calculate the difference between 2 bitmaps?



$$C(i,j) = \text{abs}(A(i,j) - B(i,j))$$

Lecture 21

48

```
% A and B have same size
[nr,nc]= size(A);
B= zeros(nr,nc);
for r= 1:nr
    for c= 1:nc
        C(r,c)= abs(A(r,c)-B(r,c));
    end
end
```

```
% A and B have same size
C= abs(A-B);
```

C is a 0-1 matrix where 1 indicates that A(i,j) and B(i,j) are different.

Section exercise

If you have an extra-long-life light bulb for your 7-by-5 display board, at which position would you install this light bulb?

Lecture 21

50