HKUSTx: ELEC1200.3x A System View of Communications: From Signals to Packets (Part 3)
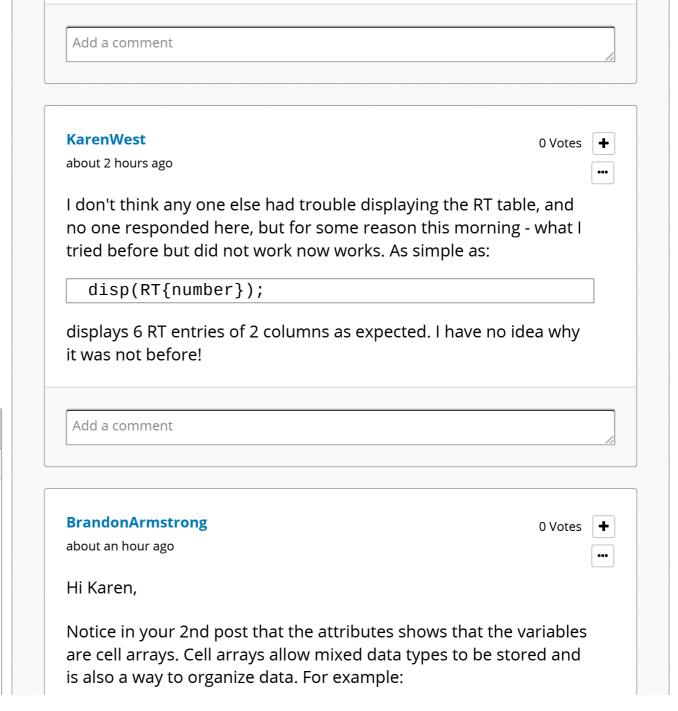
how do you display the routing table? (RT)

0 Votes

discussion posted about 18 hours ago by **KarenWest**

I've just started lab 2, and wanted to peek at the routing table. How do you display it? When I tried:

```
    disp(RT(pcount));
```

It said "6x2 double".

When I tried adding a column, I got syntax errors, such as:

```
 disp(RT(pcount,1));
 disp(RT(pcount, 2));

Incorrect syntax!  I did that when I saw the first error message,
"6x2 double matrix".
```

Thanks!

Related to: Topic 4: Routing / 4.5a Lab 2: Network Layer

This post is visible to everyone.

5 responses

↩ **Add a Response**

KarenWest   0 Votes   **+**

about 17 hours ago   **...**

I tried this and it gave:

```
        % load the routing table
        load RT.mat
        whos('-file','RT.mat')
```

gave output: (still not enough detail)

```
        Name    Size      Bytes    Class
Attributes

        RT         6x1      1248        cell

        costs      6x1      832         cell

        neighbors  6x1      832         cell
```

Add a comment

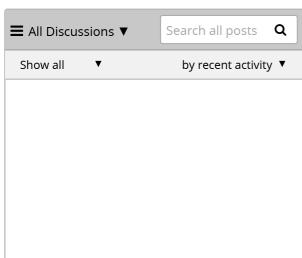**KarenWest**   0 Votes   **+**

about 5 hours ago

I was a bit shocked to wake this morning to find that no one had
responded to my question! Wow. Can anyone help with my MATLAB
syntax question, so that I can proceed to answer the conceptual
question about routing and forwarding in Lab 2? I appreciate any
help, given I've done quite a bit of web searching and trials of this
on my end.

```
load RT.mat
S = whos('-file','RT.mat');
for k = 1:length(S)
        disp(S(k));
end
for k = 1:length(S)
         disp(S(k).name);
         disp(S(k).size);
end
```

Output:

```
name: 'RT'
size: [6 1]
bytes: 1248
class: 'cell'
global: 0
sparse: 0
complex: 0
nesting: [1x1 struct]
persistent: 0

name: 'costs'
size: [6 1]
bytes: 832
class: 'cell'
global: 0
sparse: 0
complex: 0
nesting: [1x1 struct]
persistent: 0

name: 'neighbors'
size: [6 1]
bytes: 832
class: 'cell'
global: 0
sparse: 0
complex: 0
nesting: [1x1 struct]
persistent: 0
```

```
RT
6 1


costs
6 1


neighbors
6 1
```

Add a comment

KarenWest   0 Votes   **+**

about 5 hours ago   **...**

No MATLAB Syntax error for this hard coded disp:

```
    disp(RT{2}(1,2));
```

Gives you 2 as output

Maybe I can work it out from here, although I'm curious why the commands I was trying to use above did not work. I do not know MATLAB other than what I've done in parts 1 and 2 of this class.

I'm guessing I use pcount in the curly brackets, and then the (1,2) specified with other select variables, in order to see the node routed to.

Add a comment

KarenWest

0 Votes **+**

about 2 hours ago

...

I don't think any one else had trouble displaying the RT table, and no one responded here, but for some reason this morning - what I tried before but did not work now works. As simple as:

```
disp(RT{number});
```

displays 6 RT entries of 2 columns as expected. I have no idea why it was not before!

Add a comment

☰ All Discussions ▼

Search all posts 🔍

Show all ▼   by recent activity ▼

BrandonArmstrong

0 Votes **+**

about an hour ago

...

Hi Karen,

Notice in your 2nd post that the attributes shows that the variables are cell arrays. Cell arrays allow mixed data types to be stored and is also a way to organize data. For example:

```
testCell = cell(4,1);
testCell{1} = 'Brandon';
testCell{2} = 34;
testCell{3} = rand(5)
testCell{4} = testCell(1:3)
```

I now have a single variable with a character array, a scalar, a matrix, and another cell array! You've noticed the curly braces already. When you access an element of a cell array with parenthesis, you return another cell array. For example:

thirdElement = testCell(3)

returns a 1x1 cell array, not the 5x5 matrix. We've only extracted 1 cell from our 4x1 cell array. To return the contents of a cell, we must use curly braces

```
thirdElement = testCell{3}
```
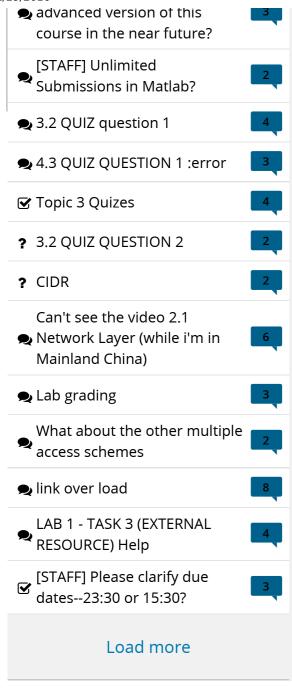
Now, thirdElement is a 5x5 matrix I can index the matrix as I before. If I want the 2nd row and 3rd column of this matrix, I do
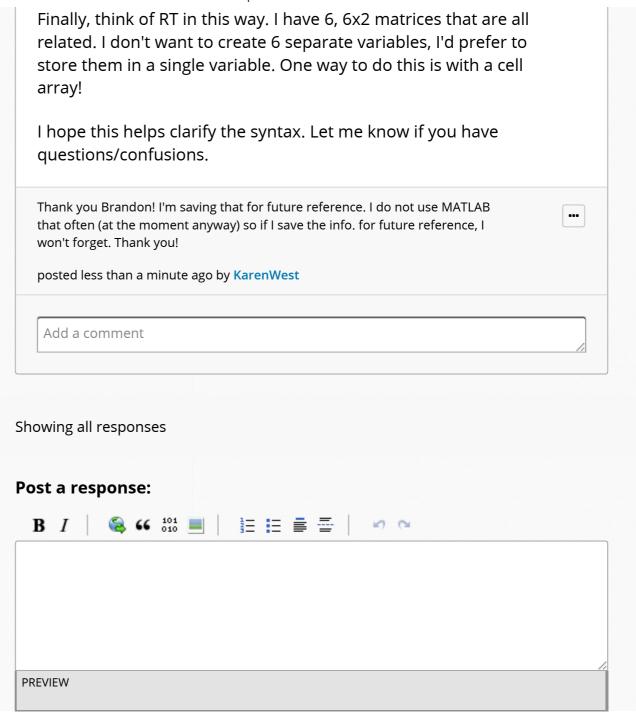
```
value = thirdElement(2,3)
```

What you discovered is a shorter way to do the same thing. Your code

```
RT{2}(1,2)
```

first grabs the contents of the 2nd cell. The contents are a 6x2 matrix (all cells in RT contain a 6x2 matrix). Then you access the 1st row and 2nd column of that matrix.

Load more

Finally, think of RT in this way. I have 6, 6x2 matrices that are all related. I don't want to create 6 separate variables, I'd prefer to store them in a single variable. One way to do this is with a cell array!

I hope this helps clarify the syntax. Let me know if you have questions/confusions.

Thank you Brandon! I'm saving that for future reference. I do not use MATLAB that often (at the moment anyway) so if I save the info. for future reference, I won't forget. Thank you!

posted less than a minute ago by **KarenWest**

Add a comment

Showing all responses

Post a response:

B *I*   🌐 66 ¹⁰¹₀₁₀ 🖼   ☰ ☰ ☰ ☰   ↩ ↪

PREVIEW

Submit

POWERED BY
OPENedX