CS 1115: Prelim 2 Solution Guide

November 8, 2012

(Name)

(Cornell ID)

Problem 1	20 points	
Problem 2	20 points	
Problem 3	20 points	
Problem 4	20 points	
Problem 5	20 points	

Average = 86 (Prob 1 = 18, Prob 2 = 16, Prob 3 = 16, Prob 4 = 17, Prob 5 = 19)
Rough Grade equivalents A = 90–100, B = 75-85, C = 60-70

1. Complete the following function so that it performs as specified

```
  function [alfa,i,j] = MaxOfAll(C)
% C is a cell array of vectors C{1},...,C{n}
% alfa is the largest numerical value that can be found in C.
% i and j are indices with the property that alfa is the value of C{i}(j)
```

Thus, if

```
C = { [ 10 -50 20], [13 5], [40 10 18 6]}
[alfa,i,j] = MaxOfAll(C)
```

then the value of `alfa`, `i`, and `j` would be 40, 3, and 1 respectively. For full credit your solution should make effective use of the `max` function[1].

Solution.

```
n = length(C)
v = zeros(n,1);
idx = zeros(n,1);
for k=1:n
  [v(k) idx(k)] = max(C{k});
end
[alfa,i] = max(v);
j = idx(i);
```

Or

```
n = length(C);
[alfa,j] = max(C{1});
i = 1;
for k=2:n
  [s mu] = max(C{k});
  if s>alfa
     alfa = s; i = k; j = mu;
  end
end
```

-5 if you have to re-scan `C{i}`.

[1] For your information, if `[beta,idx] = max([-20 30 10 15])`, then the value of `beta` is 30 and the value of `idx` is 2.

2. Assume the availability of the following functions

```
  function P = MakePoint(x,y)
% P is a point structure.
% P.x and P.y are assigned the coordinate values x and y.

  function d = Dist(P1,P2)
% P1 and P2 are points and d is the distance between them

  function E = MakeOrbit(P,A,phi,N)
% P, A, and phi are the perihelion, aphelion, and tilt (degrees) of
%    a rotated Kepler Orbit
% N is a positive integer (>=2) that specifies the number of points that
%    will make up the discrete rotated orbit.
% E is a 4-field structure that represents a discrete version of the orbit
% E.Q is a length N structure array whose components are points on the orbit
% E.s is the string-value of the orbit.
% E.F1 is a point that identifies the location of the Sun, i.e., (0,0)
% E.F2 is a point that identifies the location of the phantom focus.
```

Recall that a point is inside an ellipse if the sum of its distances to the two ellipse foci is less than the ellipse string value. Complete the following function so that it performs as specified

```
  function alfa = Inside(E1,E2)
% E1 and E2 are discrete orbits
% alfa is 1 if all points on E1 are inside the rotated Kepler orbit associated with E2.
% Otherwise, alfa is 0.
```

Efficiency matters.

Solution:

```
alfa = 1;
k = 0;
while k<length(E1.Q) && alfa
   k = k+1;
   if Dist(E1.Q(k),E2.F1) + Dist(E1.Q(k),E2.F2) >= E2.s
      alfa = 0;
   end
end
```

-8 for solutions that keep iterating even after an outside point is discovered.

3. The `uint8` range can be broken into four *quartile intervals*: $Q_1 = [0,\ 63]$, $Q_2 = [64,\ 127]$, $Q_3 = [128,\ 191]$, and $Q_4 = [192,\ 255]$. (Note that it is easy to determine the quartile interval of a `uint8` value by using `ceil`.) Complete the following function so that it performs as specified.

```
function V = Profile(fName)
% fName is the name of a .jpg file in the current directory that encodes an image.
% V is a 4-by-4-by-4 array with the property that V(i,j,k) is the
% number of pixels in the image with the property that
%            its red   value is in the i-th quartile        (1<=i<=4)
%            its green value is in the j-th quartile        (1<=j<=4)
%            its blue  value is in the k-th quartile        (1<=k<=4)

  A = imread([fName '.jpg']);
        :
        :
```

Solution.

```
[m,n,p] = size(A);
V = zeros(4,4,4);
for i=1:m
   for j= 1:n
       for k=1:3
          u = ceil(A(i,j,1)/4);
          v = ceil(A(i,j,2)/4);
          w = ceil(A(i,j,3)/4);
          V(u,v,w) = V(u,v,w) + 1;
       end
   end
end
```

4. The `sort` function[2] can be used to identify the median value in a vector:

```
[y,idx] = sort(x);
n = length(x);
medIndex = idx(ceil(n/2));
medianValue = x(medIndex);
```

We assume here and throughout this problem that n is odd. Complete the following function so that it performs as specified:

```
function [xBelow,xAbove] = Split(x)
% x is a length-n column vector with distinct values.
% n is odd and n>=3.
% xBelow is a sorted column vector of all values in x that are strictly less
%         than the median of x.
% xAbove is a sorted column vector of all values in x that are strictly larger
%         than the median of x.
```

Thus, if `x = [30; 70; 10; 20; 40; 60; 50]`, then `xBelow = [10; 20; 30]` and `xAbove = [ 50; 60; 70]`. For full credit, your solution must NOT be vectorized.

Solution.

```
[y,idx] = sort(x);
m = floor(length(x)/2);
xBelow = zeros(m,1);
xAbove = zeros(m,1);
for k=1:m
    xAbelow(k) = y(k)
    xAbove(k) =  y(m+k+1);
end
```

Up to -10 if you confuse `k` and `idx(k)`.

[2]For your information, if `[y,idx] = sort([20 40 10 30]` then `y = [10 20 30 40]` and `idx = [3 1 4 2]`.

5.(a) Assume that `n` is a positive integer and that we have the following length-60 string of amino acid mnemonics:

```
Names = 'AlaArgAsnAspCysGluGlnGlyHisIleLeuLysMetPheProSerThrTrpTyrVal';
```

(a) Write a fragment that generates a random length-n protein. Represent the protein in a length-n cell array `C`. Make effective use of `strcmp`[3].

(b) Write a fragment that generates a random length-n protein with the property that adjacent amino acids are different. Represent the protein in a length-n cell array. Thus, if `n = 4` then

```
C = { 'Cys', 'Phe', 'Cys', 'Met'}
```

is OK but

```
C = { 'Cys', 'Phe', 'Phe', 'Met'}
```

is not.

Solution.

```
C = cell(n,1);
for k=1:n
   j = ceil(20*rand());
   C{k} = Names(3*j-2:3*j);
end
```

```
C = cell(n,1)
k = 0;
while k<n
   j = ceil(20*rand())
   Next = Names(3*j-2:3*j);
   if k==0 || ~strcmp(Next,C{k})
       k = k+1;
       C{k} = Next;
   end
end
```

[3]Recall that `strcmp(s1,s2)` is 1 if the strings `s1` and `s2` are identical and is 0 otherwise.