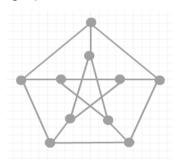
Problem Set 1

1. Mathematical calculations

(a) i. Here is the 3-regular graph:



ii. If graph G contains C₃, that means there will be a cycle which is triangle in G. Obviously, there is no triangle in this graph. And also, any vertex in this graph has three neighbor vertices because of 3-degree, however, none of three neighbors are connected, so G contains no cycle C₃.

This is a ten-vertex 4-regular graph, and it connects 2 subsets by 5 chord. If any chord connects two vertices at distance 2 or 3, then this graph would have C₃ or C₄. However, this situation does not satisfy the conditions in the definition of G.

If there is C_4 in this graph, there will exist two nodes sharing the same two neighbor vertices other than themselves. Graph G has no such two vertices, so G contains no cycle C_4 .

iii. If there is a C_{10} , then the graph consists of C plus five chords. If each chord joins vertices opposite on C, then there is a 4-cycle. Hence some chord joins vertices at distance 4 along C. Now no chord incident to a vertex opposite an endpoint of chord on C can be added without creating a cycle with at most four vertices. Therefore, this graph does not contain C_{10} .

(b)
$$A - \lambda I = \begin{bmatrix} 3 - \lambda & 4 & -1 \\ -1 & -2 - \lambda & 1 \\ 3 & 9 & -\lambda \end{bmatrix}$$

And $det(A - \lambda I) = 0$, so we get:

$$(3 - \lambda)[(-\lambda)(-2 - \lambda)] - 4(\lambda - 3) + (-1)[-9 - 3(-2 - \lambda)] = 0$$
$$-\lambda^3 + \lambda^2 + 8\lambda - 12 = 0$$
$$(\lambda - 2)(-\lambda + 2)(\lambda + 3) = 0$$

So the eigenvalues for matrix A is 2, -3, and the eigenvectors for $\lambda = 2$ is

$$\begin{pmatrix} -\frac{1}{3} \\ \frac{1}{3} \\ 1 \end{pmatrix}, \text{ the eigenvectors for } \lambda = -3 \text{ is } \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ 1 \end{pmatrix}.$$

Because the eigenvalues for matrix A is not all positive, A is not positive definite.

2. Programming

Whether graph G is bipartite: True
(D:\python\anaconda3) C:\Users\luna>_

I have 10 bipartite examples and 10 non-bipartite examples, and my examples for both bipartite and non-bipartite have various size from small to 1000. Files with "bi*.txt" are bipartite examples, and files with "non*.txt" are non-bipartite examples. "bi9.txt" and "non9.txt" are both have size 1000 nodes.

Here is the result for bipartite graphs:

```
(D:\python\anaconda3) C:\Users\luna>cd D:\CS591-GraphThoery\PS1\bipartite.py D:\CS591-GraphThoery\PS1\bip.txt Whether graph G is bipartite: True

(D:\python\anaconda3) C:\Users\luna>python D:\CS591-GraphThoery\PS1\bipartite.py D:\CS591-GraphThoery\PS1\bio.txt Whether graph G is bipartite: True

(D:\python\anaconda3) C:\Users\luna>python D:\CS591-GraphThoery\PS1\bipartite.py D:\CS591-GraphThoery\PS1\bilo.txt Whether graph G is bipartite: True

(D:\python\anaconda3) C:\Users\luna>python D:\CS591-GraphThoery\PS1\bipartite.py D:\CS591-GraphThoery\PS1\bipartite.p
```

(D:\python\anaconda3) C:\Users\luna>python D:\CS591-GraphThoery\PS1\bipartite.py D:\CS591-GraphThoery\PS1\bi8.txt

Here is the result for non-bipartite graphs:

Anaconda Prompt

 $\begin{tabular}{ll} $(D:\python\anaconda3) $C:\Users\luna\python $D:\CS591$-GraphThoery\PS1\bipartite.py $D:\CS591$-GraphThoery\PS1\non0.txt $Whether graph G is bipartite: False $$(D:\python\anaconda3) $(D:\python\anaconda3) $(D:\python\anaconda3)$

(D:\python\anaconda3) C:\Users\luna>python D:\CS591-GraphThoery\PS1\bipartite.py D:\CS591-GraphThoery\PS1\non1.txt Whether graph G is bipartite: False

 $(D:\python\anaconda3) C:\Users\luna\python D:\CS591-GraphThoery\PS1\bipartite.py D:\CS591-GraphThoery\PS1\non2.txt Whether graph G is bipartite: False$

(D:\python\anaconda3) C:\Users\luna>python D:\CS591-GraphThoery\PS1\bipartite.py D:\CS591-GraphThoery\PS1\non3.txt Whether graph G is bipartite: False

 $\begin{tabular}{ll} $(D:\p thon\anaconda3) $C:\Users\luna>python $D:\CS591$-GraphThoery\PS1\bipartite.py $D:\CS591$-GraphThoery\PS1\non4.txt $Whether graph G is bipartite: False $$(D:\p thought between the property of the property of$

 $\begin{tabular}{ll} $(D:\p thon\anaconda3) $C:\Users\luna>python $D:\CS591-GraphThoery\PS1\bipartite.py $D:\CS591-GraphThoery\PS1\non5.txt $Whether graph G is bipartite: False $$(D:\p thought for the property of the p$

 $\begin{tabular}{ll} $(D:\p thon\anaconda3) $C:\Users\luna>python $D:\CS591-GraphThoery\PS1\bipartite.py $D:\CS591-GraphThoery\PS1\non6.txt $Whether graph G is bipartite: False $$(D:\p thought for the property of the p$

 $\begin{tabular}{ll} $(D: \python\anaconda3) $C: \Users\luna\python $D: \CS591-GraphThoery\PS1\begin{tabular}{ll} PS1\begin{tabular}{ll} PS1\begin{tabular}{ll$

 $\begin{tabular}{ll} $(D:\p thon \an aconda3) $C:\Users\luna>python $D:\CS591-GraphThoery\PS1\begin{tabular}{ll} $D:\CS591-GraphThoery\PS1\begin{tabular}{$

(D:\python\anaconda3) C:\Users\luna>_