Please implement graph data structures below using file reading/writing method to input/output:

1) Adj Matrix (general case)

4

0 1 1 0

1 0 1 1

1 1 0 0

0 1 0 0

Advance requirements:

a) calculate degree of vertices (duyệt theo dòng, tính tổng. Tổng của dòng i cho biết bậc của đỉnh thứ i)

b) number of edges (need to check if the graph is directed or not)

//Kiểm tra đồ thị có hướng

Duyệt đồ thị, kiểm tra từng cặp phần tử đối xứng nhau, nếu tồn tại một cặp a[i][j] != a[j][i] bất kỳ thì đồ thị có hướng

// tính tổng cạnh của đồ thị

Duyệt đồ thị, nếu a[i][j]!=0 thì tổng\_cạnh ++;

Nếu đồ thị vô hướng thì tổng\_cạnh = tổng\_cạnh/2;

c) tranfer adj matrix into adj list/edge list

Duration: 45 minutes

Due to 11:20

2) Adj List (general case)

4

1 2

2 3 4

3 2

4 2

Advance requirements:

a) cal degree of vertices

b) number of edges

c) tranfer adj list into adj matrix...

3) Edge list

1 1 2

1 1 3

1 2 1

1 2 3

1 2 4

1 3 1

1 3 2

1 4 2

advance requirements:

a) number of vertices

b) number of edge

c) degree of vertices

d) tranfer edge list into adj matrix...

for (int i=0, i<V, i++)

{

temp = adj[i].head;

while (temp.next!=null)

adjMat[i][temp.key]=1;

}

covert adjMat to adjList --undirected graph

for (int i=0; i<v;i++)

for(j=i+1; j<v; j++)

if (adjMat[i][j] == 1) adjListArray.get(i).add(j);