**Problem 1**

Suppose G has edge e = {a,b} that does not belong to T

If T is a DFS tree, by the property of DFS tree. One of the two ends must be on ancestor of the other.

Assume a is ancestor of b

If T is a BFS tree, by the property of BFS tree, the two nodes a and b can be differed by distance at most 1

But if a is ancestor of b and the property of BFS tree holds

Then a must be the direct parent of b

This means that {a,b} is an edge in T Thus proved by contradiction

**Problem 2**

Read the input using buffered Reader.

Create a adjacency list and read the output to fill in the Edges.

Queue q //Use a queue

Boolean visited[] //all initialized to be false

dist[] //shortest distance

path[] //# of path to be returned, initialize to 0

count

enqueue(source)

While Q is not empty

v = dequeue(Q)

For each g adjacent to v

if visited[g] = false

enqueue(g)

Then visited[g] = true;

If(g.dist > v.dist+1)

g.dist = v.dist +1

g.path = v.path

if(g.dist = v.dist +1)

g.path = g.path + v.path

return path