In [34]:

```
import random
import queue
def bfs(start, graph, depth):
    q = queue.Queue()
    nVertices = 0
    visited = {}
    q.put(start)
    visited[start] = 1
    while not q.empty():
        if depth < 0:
            break
        size = q.qsize()
        nVertices += size
        for i in range(size):
            vertex = q.get()
            for j in graph[vertex]:
                if not visited.get(j,0):
                    q.put(j)
                    visited[j] = 1
        depth = depth - 1
    return  nVertices

def compute_degree(d, n):
    largest = 0
    average = 0
    vertex = -1
    for i in list(d.keys()):
        length = len(d[i])
        average+= length
        if largest < length:
            largest = length
            vertex = i
    average = average/n
    return vertex,largest,average
def populate(d, u, v):
    adjListU = d.get(u, [])
    adjListU.append(v)
    d[u] = adjListU

d = {}
with open("C:\\Users\\pramod\\Desktop\\roadNet-TX.txt") as fp:
    line = fp.readline()
    while line:
        line = line.strip().split()
        u,v=int(line[0]),int(line[1])
        populate(d, u, v)
        line = fp.readline()
n = 1393382
vertex, largest, average = compute_degree(d, n)
print("vertex "+str(vertex)+"has highest degree "+str(largest)+" and average degree is:"+str(average))
random_vertices = random.sample(range(1, n+1), 10)
depth = 50
for i in random_vertices:
    number_vertices = bfs(i, d , depth)
    print("number of vertices at depth "+str(depth) + " from " +str(i)+" is "+str(number_vertices))
```