Assignment 3, Due Date 11.13.2017

In [285]:

```
import networkx as nx
import matplotlib.pyplot as plt
import numpy as np
import warnings
from datetime import datetime
from collections import Counter
%matplotlib inline
```

In [286]:

```
social = nx.read_gpickle('linkedin.gpickle')
```

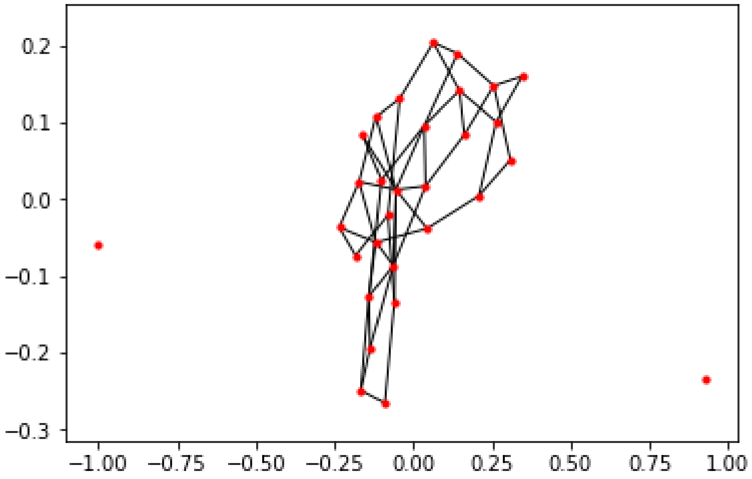
In [287]:

```
print(nx.info(social))
```

```
Name: 
Type: Graph
Number of nodes: 30
Number of edges: 48
Average degree:   3.2000
```

In [288]:

```
nx.draw_networkx(social,node_color='r', node_size=10, with_labels=False)
```



```
In [289]: #get the node attributes
          social.nodes(data=True)
```

```
Out[289]: NodeDataView({0: {'age': 20, 'sex': 'Male'}, 1: {'age': 21, 'sex': 'Femal
          e'}, 2: {'age': 19, 'sex': 'Male'}, 3: {'age': 29, 'sex': 'Female'}, 4:
          {'age': 30, 'sex': 'Male'}, 5: {'age': 26, 'sex': 'Female'}, 6: {'age': 2
          1, 'sex': 'Male'}, 7: {'age': 17, 'sex': 'Female'}, 8: {'age': 21, 'sex':
          'Male'}, 9: {'age': 14, 'sex': 'Male'}, 10: {'age': 23, 'sex': 'Male'}, 1
          1: {'age': 17, 'sex': 'Female'}, 12: {'age': 19, 'sex': 'Male'}, 13: {'ag
          e': 27, 'sex': 'Female'}, 14: {'age': 29, 'sex': 'Female'}, 15: {'age': 1
          4, 'sex': 'Male'}, 16: {'age': 18, 'sex': 'Female'}, 17: {'age': 21, 'se
          x': 'Female'}, 18: {'age': 19, 'sex': 'Male'}, 19: {'age': 19, 'sex': 'Fe
          male'}, 20: {'age': 19, 'sex': 'Female'}, 21: {'age': 21, 'sex': 'Male'},
          22: {'age': 30, 'sex': 'Female'}, 23: {'age': 25, 'sex': 'Female'}, 24:
          {'age': 13, 'sex': 'Male'}, 25: {'age': 24, 'sex': 'Female'}, 26: {'age':
          23, 'sex': 'Male'}, 27: {'age': 21, 'sex': 'Male'}, 28: {'age': 29, 'se
          x': 'Female'}, 29: {'age': 25, 'sex': 'Male'}})
```

Question 1: Write a program that presents the range of dates (earliest and last dates) during which these relationships were forged?

```
In [290]: Q1 = social.edges(data=True)

          #indexing the 3rd element of the list to strip out the date
          xx = [x[2] for x in social.edges(data="date")]
          print("Earliest Date is", max(xx))
          print("Latest Date is" ,min(xx))
```

```
Earliest Date is 2011-11-04 00:00:00
Latest Date is 2002-05-20 00:00:00
```

Question 2: Write a program that demonstrates if node 5 and 25 are friends (directly or indirectly)

```
In [291]: #finding the if 5 has any edges
          #if node 5 doesn't have any edges then it's not connected to any nodes
          print("Edges:", nx.edges(social,5))
          print("All Neighbors:",list(nx.all_neighbors(social,5)))
          print("Neighbors:",list(social.neighbors(5)))
          print("Node 5 does not have any friends, directly or indirectly")
```

```
Edges: []
All Neighbors: []
Neighbors: []
Node 5 does not have any friends, directly or indirectly
```

Question 3: Write a program that lists direct

friends of node 4

In [292]:

```
# printing the info for node 4
print(nx.info(social,4))

print("Neighbors V2",list(social.neighbors(4)))

se = nx.edges(social,4)

c = [el[1] for el in se]
c.append(4)

print("Direct Friends of Node 4 :",c)
```

```
Node 4 has the following properties:
Degree: 3
Neighbors: 1 19 28
Neighbors V2 [1, 19, 28]
Direct Friends of Node 4 : [1, 19, 28, 4]
```

Question 4: Write a program that presents the most popular person

In [293]:

```
dc =nx.degree_centrality(social)
#print (dc)
print("The most popular person is Node", max(zip(dc.values(), dc.keys()))[1]
```

```
The most popular person is Node 19
```