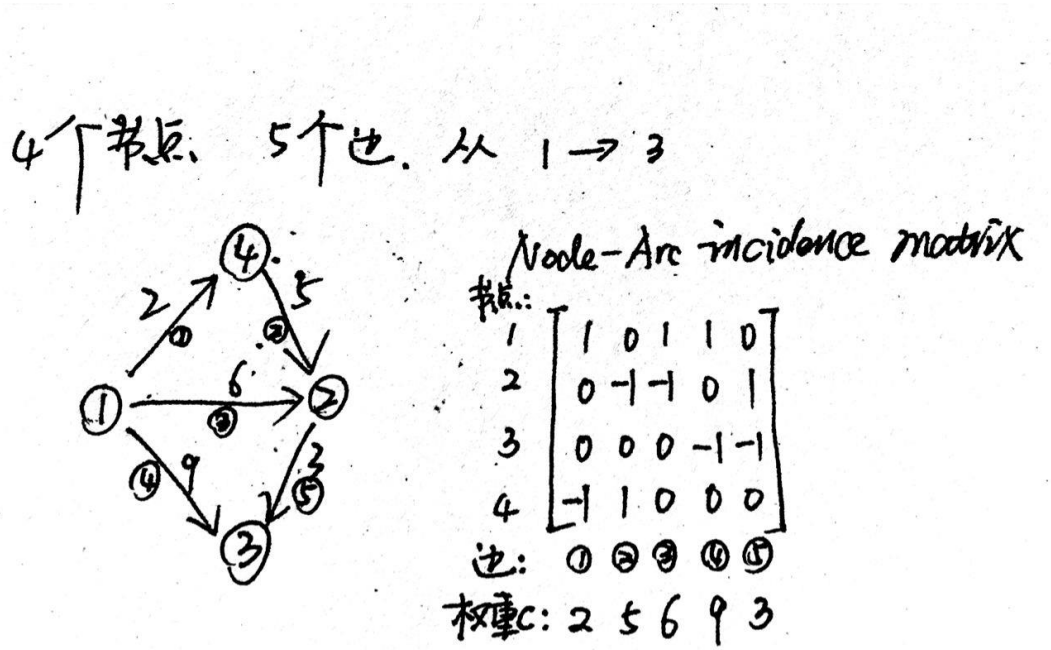
Dijkstra 算法：



步骤：

创建图

```
#encoding:utf-8
from calcpath import *
import networkx as nx

if __name__ == '__main__':
    #Create Direct Graph
    DG = nx.DiGraph([(1,2),
                     (1,4),
                     (1,3),
                     (2,3),
                     (4,2)])
    #Add weight to edges
    DG[1][2]['weight'] = 6
    DG[1][4]['weight'] = 2
    DG[1][3]['weight'] = 9
    DG[2][3]['weight'] = 3
    DG[4][2]['weight'] = 5
```

```
#Add capability to edges
DG[1][2]['capability'] = 1
DG[1][4]['capability'] = 1
DG[1][3]['capability'] = 1
DG[2][3]['capability'] = 1
DG[4][2]['capability'] = 1
#dijkstra calc
tmp = PathCalc(G = DG,
               start_point = 1,
               end_point = 3)
```

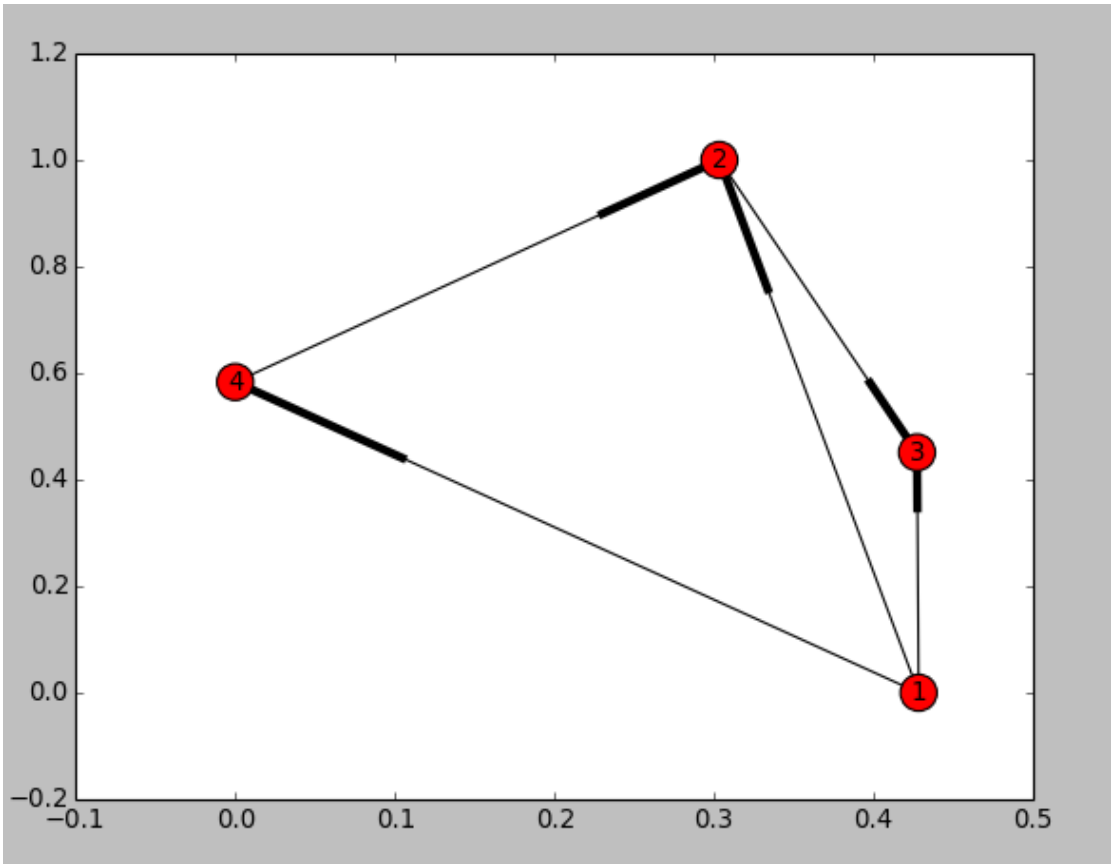
计算:

```
    #dijkstra calc
    tmp = PathCalc(G = DG,
                   start_point = 1,
                   end_point = 3)
    start = int(time.time()*1000)
    for i in range(1000):
        tmp.dijkstra_calc(field='weight',capability_min=0)
    stop = int(time.time()*1000)
    print 'dial 1000 times cost %d ms' % (stop - start)
    print 'the route is ',tmp.get_path()
    print 'the min of weight',tmp.get_weight()

    nx.drawing.draw_networkx(DG)

    ply.show()
    print tmp.get_distance()
```

运行结果：

```
dial 1000 times cost 57 ms
the route is  [1, 2, 3]
the min of weight 9
```

电脑作图：有向边的箭头用途中黑色加粗部分代替


1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
1
2
3
4