


```

# find the longest of the shortest paths
def findLongestOfTheShortestPaths(graph):
    longest = 0
    path = None
    for node in graph:
        for node2 in graph:
            shortpath = findShortestPath(graph,node,node2)
            if shortpath != None:
                if len(shortpath) > longest:
                    path = shortpath
                    longest = len(shortpath)
    return path

# find the average distance between two pages
def findAverageDistance(graph):
    length = 0.0
    count = 0.0
    for node in graph:
        for node2 in graph:
            path = findShortestPath(graph,node,node2)
            if path != None:
                length = length + len(path)
                count = count + 1
    length = length / count

    return length

#finding the shortest cycle from a page and back to the same page
def findShortestCycle(graph, start, path=[]):
    shortest = 0
    if not graph.has_key(start):
        return path
    for node in graph[start]:
        try:
            if (len(path) < len(findShortestPath(graph,node,start))) and (len(path) >= 0):
                path = findShortestPath(graph,node,start)
        except Exception, e:
            path = path
    return path

# find the average distance from a page to the page again
def averageCycleLength(graph):
    totaldistance = 0.0
    count = 0.0
    for node in graph:
        shortpath = findShortestCycle(graph, node)
        if shortpath != None:
            totaldistance = totaldistance + len(shortpath)
            count = count + 1
    return totaldistance / count

## does not work
def findAverageNumberOfLinks(graph):
    count = 0.0
    for node in graph:
        count = count + len(graph[node])
    count = count / len(graph)
    return count

```