Assignment4：实现PageRank算法的Google快速版

作业要求

1. 实现PageRank算法的Google快速版，在给定的具有7115个节点、103689条边的有向图（存储在dataset.txt），完成算法测试
2. 计算得到每一个node的PageRank score

作业过程

一、 实现PageRank算法

主要功能在于：

1. 读文件，初始化encode sparse matrix: M,迭代PageRank使用的r^{new}, r^{old}
2. 计算$r^{new} = (\beta M + (1-\beta)/N)r^{old}$迭代$r^{new}$和$r^{old}$，直至$|r^{new} - r^{old}| < e$

代码实现：

1. 初始化变量

将文件中所有出现的节点初始化于M、r^{new}、r^{old}中

```
M = {}
R_new, R_old = {},{}
with open('dataset.txt', 'r') as data_file:
        for line in data_file:
            if line[0] == '#': continue
            key, value = line.split()
            if key not in M:
                M[key] = [value]
                R_old[key] = 1/num
                R_new[key] = (1-beta)/num
            else:
                M[key].append(value)
            if value not in M:
                M[value] = []
                R_old[value] = 1/num
                R_new[value] = (1-beta)/num
```

2. 循环迭代r^{new}、r^{old}至收敛

```
    epoch = 0
    while epoch < 500:
        # initialize R_new
        for i in R_new:
            R_new[i] = (1-beta)/num
        # calculate R_new
        for i in R_old:
            d_i = len(M[i])
            for dest in M[i]:
                R_new[dest] += beta*(R_old[i]/d_i)
        # calculate difference
```

```
        diff_sum = 0
        for i in R_old:
            diff_sum += np.abs(R_old[i] - R_new[i])
            R_old[i] = R_new[i]
        print(f"the diff in epoch {epoch} is {diff_sum}")
        # determine whether to break the loop
        if diff_sum < e: break
        epoch += 1
```

3. 输出结果

将结果写入文件保存

```
with open('result.txt', 'w') as file:
        file.write(f"page\t rank\n")
        for key, values in R_new.items():
            file.write(f"{key}\t {values}\n")
```

二、 运行测试

设置参数

```
num = 7115  # num of nodes
beta = 0.8  # probability of follow a link
e = 10e-50  # error bound
```

运行结果

```
the diff in epoch 73 is 2.236166980751353e-19
the diff in epoch 74 is 3.6591823321385775e-19
the diff in epoch 75 is 2.507217523872729e-19
the diff in epoch 76 is 4.87890977618477e-19
the diff in epoch 77 is 2.439454888092385e-19
the diff in epoch 78 is 1.7618285302889447e-19
the diff in epoch 79 is 2.710505431213761e-20
the diff in epoch 80 is 0.0
```

result.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

page	rank
30	8.727114373537817e-05
1412	0.00040686444562497453
3352	0.0008635928404331848
5254	0.0010511387585008419
5543	0.000513994934985818
7478	0.0004086583018997373
3	0.00010202629877110322
28	0.0008422913268065434
39	0.00017710622765532548
54	0.00017165428123957368
108	0.0002158762722549801
152	0.0002875559411411694

总结

本次作业简单实现了PageRank算法，在课上对PageRank伪代码了解后总体写下来比较顺畅，主要熟悉了算法的代码实现，实现过程通过构造课件上的例子比对PageRank的结果来确保代码的正确性。