گزارش مستندات تمرین پنجم

درس مبانی بازیابی اطلاعات و جستجوی وب

ارائه دهندگان:

سحر محمدی - فائزه سید موسوی - محمد ملائی - صالح شیروانی

دانشگاه فردوسي مشهد - دانشکده مهندسي -گروه کامپیوتر

بهار 1402

اعضای گروه و درصد مشارکت

سحر محمدی	فائزه سید موسوی	محمد ملائی	صالح شیروانی
۱۰	۱۰	۱۰	۱۰

توضیحات

در این تمرین قصد داریم مقدار pagerank را برای گره‌های گراف ورودی محاسبه کنیم. نیاز به پیاده‌سازی بخش های پایین داریم:

۱- خواندن ورودی‌ها از فایل

خط اول فایل ورودی برنامه، تعداد گراف هایی را نشان می‌دهد که قصد داریم مقدار pagerank را برای هر گره آن به دست آوریم. در ادامه نیز به تعداد گراف‌ها ماتریس مجاورت قرار دارد که یک گراف را به طور کامل توصیف می‌کند.

2- ایجاد گراف از روی ماتریس مجاورت

پس از خواندن مقادیر ورودی از فایل، گراف مربوط به هر یک از ماتریس‌های مجاورت دریافتی را تشکیل می‌دهیم. برای تشکیل گراف‌ها از پکیج networkx پایتون استفاده می‌کنیم.

با توجه به اینکه گراف‌هایی که در این تمرین مورد استفاده قرار می‌گیرند جهت‌دار هستند، باید از کلاس DiGraph این پکیج استفاده کنیم تا گراف جهت‌دار ایجاد کنیم.

همچنین به هر گره یک نام(node_name) اختصاص می‌دهیم. ما این نام را یک عدد در نظر گرفته ایم. این عدد در بازه‌ي 1 تا تعداد گره‌های گراف قرار می‌گیرد.

پس از تشکیل یک گراف، آن را در یک دیکشنری با نام graphs ذخیره می‌کنیم. این کار باعث می‌شود که برنامه برای هر تعداد گراف که در ورودی دریافت می‌کند پاسخگو باشد.

نکته: کلیدهای دیکشنری گراف نام گراف هستند. نام گراف یک رشته متشکل از کلمه graph و شماره گراف است. شماره گراف بر اساس ترتیب قرار گرفتن آن‌ها در فایل ورودی مشخص می‌شود. به عنوان مثال graph1 نشان دهنده اولین گرافی است که در فایل ورودی برنامه وجود دارد. مقادیر این دیکشنری هم طبیعتا شی گراف ایجاد شده است.

3- محاسبه مقدار pagerank برای هر گره از هر گراف

ابتدا یک دیکشنری به نام scores تعریف می‌کنیم. ساختار این دیکشنری به صورت زیر است:

```
scores = {
        Graph_name: {
                Iteration_number: {
                        Node_name: pagerank_value
                }
        }
}
```

از ساختار این گراف مشخص است که هدف ما این است که مقدار pagerank هر گره هر گراف را در هر iteration ذخیره کنیم.

برای محاسبه pagerank از فرمول زیر استفاده می‌کنیم:

$$PR(A) = \frac{1-d}{N} + d(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} + \ldots)$$

که در آن $PR(x)$ مقدار pagerank مربوط به صفحه x است و $L(x)$ مشخص کننده تعداد ارجاعات صفحه x به صفحات دیگر است. همچنین N تعداد کل صفحات و d فاکتور تعدیل است.

نکته‌ای که باید به آن توجه کرد این است که با توجه به تست کیسی که در اختیارمان قرار گرفت مقدار جدید محاسبه شده pagerank برای هر گره در یک iteration نباید در محاسبه pagerank گره‌های دیگر در همین iteration تاثیری داشته باشد. به بیان دیگر اگر بخواهیم pagerank گره A را در iteration شماره iام محاسبه کنیم و یک یال از گره B به گره A وجود داشته باشد، طبق الگوریتم در محاسبه pagerank گره A به مقدار pagerank گره B نیاز داریم ($PR(B)$) در فرمول بالا). اما باید از مقدار pagerank گره B در iteration شماره i-1 استفاده کنیم نه iteration شماره iام.

همچنین به این نکته هم توجه داریم که مقدار pagerank تمامی گره‌ها در iteration شماره 1 برابر با عدد یک است.

در نهایت پس از محاسبه مقدار pagerank هر گره، مقدار محاسبه شده را در دیکشنری scores ذخیره می‌کنیم.

4- نمایش خروجی

در این برنامه سه مدل خروجی ایجاد می‌کنیم:

1. فایل output.txt: مطابق صورت تمرین مقدار pagerank هر گره از هر گراف را به ازای 10 گردش(iteration) در این فایل ذخیره کرده ایم.

2. چاپ مقادیر در خروجی: در این مدل هم مقادیر pagerank محاسبه شده در 10 گردش اول را در خروجی چاپ می‌کنیم.
3. Plot گراف‌ها: در این مدل با استفاده از پکیج networx گراف‌ها را به همراه مقدار pagerank محاسبه شده برای هر گره plot می‌کنیم.