

مرتب‌سازی توپولوژیکی

پروژه پایانی درس طراحی و تحلیل الگوریتم‌ها، ترم دوم سال تحصیلی ۱۴۰۳-۱۴۰۴، دانشگاه کردستان، مدرس: نامانج خرمیان

۱. مقدمه

در علوم کامپیوتر، مرتب‌سازی توپولوژیکی یا ترتیب توپولوژیکی یک گراف جهت‌دار بدون دور، یک ترتیب خطی از رئوس آن است به طوری که به ازای هر یال جهت‌دار از رأس u به رأس v ، رأس u قبل از رأس v در ترتیب قرار گیرد. یک ترتیب توپولوژیکی تنها در صورتی امکان‌پذیر است که گراف هیچ دور جهت‌داری نداشته باشد، یعنی یک گراف جهت‌دار غیرمردور (DAG) باشد. جهت یادآوری مطلب به فصل 20.4 کتاب یا اسلایدهای ۶۵ و ۶۶ تدریس شده سر کلاس مراجعه کنید.

۲. هدف پروژه

هدف این پروژه، پیاده‌سازی الگوریتم مرتب‌سازی توپولوژیکی برای یک گراف جهت‌دار غیرمردور (DAG) است. برنامه شما به عنوان ورودی یک گراف DAG را که با ماتریس مجاورت نمایش داده شده است دریافت کرده و به عنوان خروجی یک ترتیب خطی از رئوس را ارائه می‌دهد. همانطور که سر کلاس توضیح داده شده، شما باید از الگوریتم مبتنی بر جستجوی اول عمق (DFS) برای انجام این پروژه استفاده کنید.

۳. الزامات پروژه

- زبان برنامه‌نویسی: شما می‌توانید از یکی از زبان‌های C، C++ یا Java برای این پروژه استفاده کنید.
- فرمت ورودی: ورودی برنامه شما یک فایل متنی به نام graph.txt خواهد بود. خط اول این فایل شامل یک عدد صحیح n ، یعنی تعداد رئوس گراف است. n خط بعدی، ماتریس مجاورت گراف را نشان می‌دهند. رئوس از 0 تا $n - 1$ شماره‌گذاری شده‌اند. به عنوان مثال، درایه سطر i -ام و ستون j -ام برابر با 1 خواهد بود اگر یالی از رأس i به رأس j وجود داشته باشد، و در غیر این صورت 0 خواهد بود.
- فرمت خروجی: برنامه شما باید یک خط واحد را در کنسول چاپ کند که شامل رئوس مرتب‌شده به صورت توپولوژیکی است و با فاصله از هم جدا شده‌اند. اگر گراف حاوی دور باشد، برنامه شما باید یک پیام خطا نمایش دهد.
- الگوریتم: شما باید الگوریتم مرتب‌سازی توپولوژیکی را با استفاده از جستجوی اول عمق (DFS) پیاده‌سازی کنید. ایده اصلی این است که یک پیمایش DFS روی گراف انجام دهید. در طول پیمایش، پس از اینکه تمام همسایه‌های یک رأس بازدید شدند، یعنی به زمان پایان (Finish Time) رأس رسیده‌ایم، آن رأس به ابتدای یک لیست اضافه می‌شود. این لیست در نهایت ترتیب توپولوژیکی رئوس خواهد بود.
- تشخیص دور: پیاده‌سازی شما باید قادر به تشخیص وجود دور در گراف باشد. مرتب‌سازی توپولوژیکی برای گرافی که دور دارد امکان‌پذیر نیست. اگر دوری تشخیص داده شد، برنامه شما باید آن را گزارش کرده و به درستی پایان یابد.
- مستندسازی کد: کد شما باید به خوبی با توضیحات (کامنت) مستند شده باشد و منطق پیاده‌سازی شما، به ویژه تابع DFS و تابع اصلی مرتب‌سازی توپولوژیکی، را توضیح دهد.

۴. مثال محتوای فایل graph.txt

```
6
0 1 1 0 0 0
0 0 0 1 1 0
0 0 0 0 0 1
0 0 0 0 0 0
0 0 0 1 0 0
0 0 0 1 0 0
```

```
0 1 2 4 5 3
```

خروجی در کنسول:

(توجه: ترتیب‌های توپولوژیکی معتبر دیگری نیز ممکن است وجود داشته باشند اما نیازی به نشان دادن همه آنها نیست)

۵. نحوه تحویل پروژه در گروه تلگرامی اطلاع رسانی می‌شود