به نام خدا

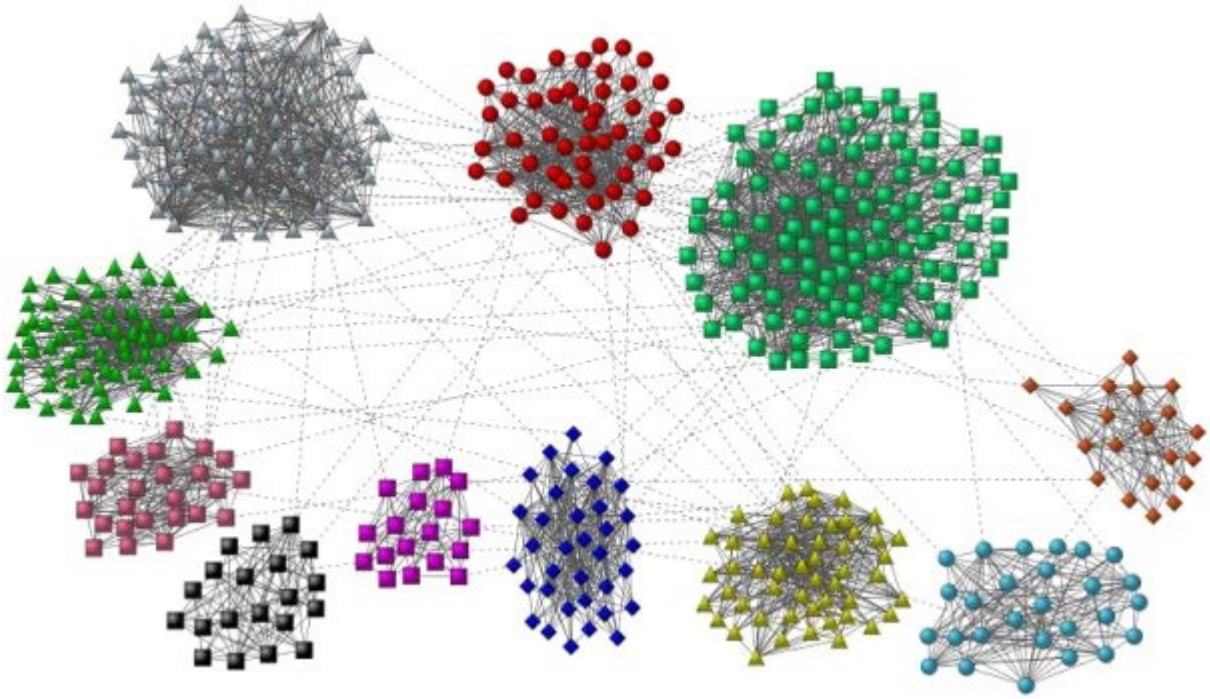
پروژه امتیازی درس طراحی الگوریتم

تشخیص اجتماعات در گراف

هدف این پروژه پیاده‌سازی و بررسی یک الگوریتم مربوط به تشخیص ساختارهای اجتماعی در گراف‌ها است. در ادامه جزئیات الگوریتم و بررسی‌های خواسته شده امده است.

اجتماع

اجتماع در یک گراف به زیرگرافی گفته میشود که تعداد یال های بین رأسهای داخلی زیر گراف نسبت به یال های خارج شده از آن زیرگراف (به رئوس دیگر موجود در گراف) بیشتر باشد. به عنوان مثال، شکل زیر اجتماع های موجود در یک گراف را نشان میدهد.



شرح الگوریتم

در این الگوریتم، با حذف یال‌های بین اجتماع ها، گراف به زیر گرافهای متراکم تر تقسیم میشود. معیاری که این الگوریتم برای امتیازدهی به یالها استفاده میکند، عبارت است از:

$$C_{ij} = \frac{Z_{ij} + 1}{min[k_i - 1, k_j - 1]}$$

که در آن z_{ij} برابر با تعداد دورهای به طول سه است که راس i و j در آن دور دیده می‌شود. بدیهی است که اگر یالی بین دو راس i و j وجود نداشته باشد مقدار C_{ij} برای آن‌ها تعریف نمی‌شود. همچنین k_i درجه‌ی راس i و k_j درجه‌ی راس j است. بنابراین، اگر درجه یکی از رأسها عدد یک باشد، آنگاه مخرج کسر برابر با صفر میشود که تعریف نشده است؛ یعنی در این حالت، مقدار C_{ij} را باید برابر با عددی بسیار بزرگ در نظر بگیرید.

قدم‌های الگوریتم به شرح زیر است:

۱. محاسبه‌ی امتیاز C_{ij} برای تمام یال‌ها.
۲. مرتب‌سازی صعودی یال ها بر اساس امتیاز محاسبه‌شده.
۳. حذف یال با کوچکترین مقدار C_{ij}از گراف.
٤. اگر گراف به دو بخش تقسیم شده است، آنگاه پایان الگوریتم.
٥. محاسبه‌ی امتیاز C_{ij} برای یال‌هایی که امتیاز آن‌ها بعد از حذف یال در قدم سوم تغییر می‌کند.(بدیهی است که امتیاز بخشی از یال‌های گراف در هر دوره اجرای الگوریتم تغییر می‌کند و امتیاز باقی یال‌های گراف ثابت می‌ماند و نیازی به محاسبه‌ی دوباره ندارد.)
٦. برو به قدم اول.

ورودی برنامه فایلی در قالب CSV است که در آن، هر راس با یک عدد طبیعی نشان داده میشود و هر سطر از آن بیانگر یک یال از گراف ورودی است. تمام فایل باید یکجا خوانده شده و در حافظه RAM بارگذاری شود.

۱. جهت ذخیره گراف در حافظه، باید از ساختمان داده لیست همسایگی استفاده کنید.
۲. جهت مرتب سازی باید از الگوریتم‌های زیر پشتیبانی شود.
   a. Insertion sort
   b. Bubble sort
   c. Merge Sort
   d. Quick Sort

اجرا:

در زمان اجرا برنامه باید فایل مورد نظر حاوی اطلاعات گراف و نام الگوریتم انتخابی برای مرتب‌سازی را دریافت کند. به عنوان مثال:

1. RUN Bubble test1.csv
2. RUN Quick test2.csv
3. RUN Merge test1.csv
4. RUN Insertion test2.csv

خروجی:

خروجی هر اجرا از برنامه باید حاوی موارد زیر باشد.

۱. زمان مصرف شده برای خواندن اطلاعات گراف از فایل csv و ذخیره‌ی آن در لیست مجاورت.
۲. زمان مصرف شده برای محاسبه‌ی اولیه امتیازها.
۳. نمودار زمان مصرف شده در قدم مرتب سازی از ابتدا تا انتهای الگوریتم.
٤. یک فایل csv که در سطرهای آن شماره‌ی هر راس به همراه اجتماع آن امده‌است.(A یا B)

نکات پیاده‌سازی:

۱. پیاده سازی باید به صورت تک نفره باشد و محدودیتی برای زبان پیاده سازی وجود ندارد. اما دقت کنید که استفاده از ساختمان داده های آماده و به صورت کتابخانه مجاز نیست.
۲. در حین انجام پروژه، بحث و بررسی بین دانشجویان آزاد است اما هر دانشجو موظف است به تنهایی پروژه را انجام دهد و در هنگام تحویل حضوری، دانشجو باید به تمام جزئیات پیاده‌سازی کد کاملاً مسلط باشد. در مورد قسمتهایی از کد و نحوه عملکرد برنامه نیز از دانشجو سوال خواهد شد. همچنین با مواردی که تقلب و کپی کردن تشخیص داده شوند، برخورد جدی خواهد شد.
۳. موعد تحویل این پروژه تا ساعت ۱۱:۵۵ روز یکشنبه ۱۹ بهمن خواهد بود. پوشه مربوط به کد پروژه را همراه با فایل pdf حاوی شرح انجام پروژه، نحوه اجرای برنامه و گزارش مربوط به تحلیل ساختمان داده های مورد استفاده و نیز فایل‌های تست را در قالب یک فایل zip به شکل زیر بارگذاری کنید. زمان و چگونگی نحوه تحویل متعاقباً اعلام میشود.

StudentNumber-FirstName-LastName-Project.zip