

بسمه تعالی

گزارش پروژه‌ی شماره‌ی ۲۲ درس طراحی الگوریتم

رنگ آمیزی کامل گراف یا Total Coloring

استاد:

جناب آقای دکتر باقری

تدریس‌یار:

جناب آقای صابر غلامی

پارسا انعامی

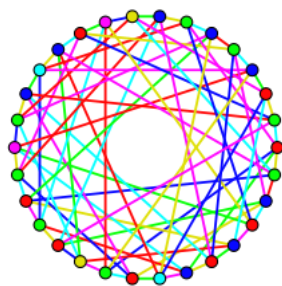
۹۵۳۱۹۰۸

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی امیرکبیر

بهار ۱۳۹۷

توضیح کلی الگوریتم و مدل رنگ آمیزی



شیوه‌ی رنگ آمیزی total coloring یکی از راه‌های رنگ آمیزی گراف می‌باشد که در آن تمام رئوس و یال‌ها رنگ آمیزی می‌شوند، به طوری که هیچ رأسی، هیچ یال متلاقی و همچنین هیچ یال و رأس‌های دو سر آن دارای رنگی مشترک نباشند.

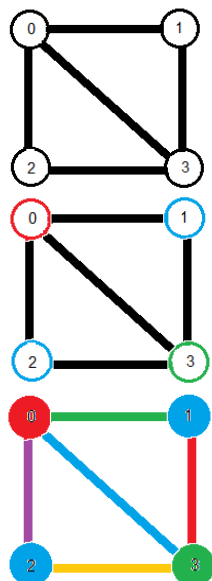
مثل گراف مقابل:

در حالت کلی این مدل رنگ آمیزی محدودیتی ندارد مگر آنکه در صورت سوال موردی ذکر شود.

الگوریتمی در این مسئله بهینه محسوب می‌شود که کمترین تعداد رنگ را برای رنگ آمیزی گراف استفاده کند.

در الگوریتمی که اینجا ارائه نموده‌ایم، ابتدا رئوس گراف را بدون توجه به یال‌ها، رنگ آمیزی می‌کنیم. سپس به رنگ کردن یال‌ها می‌پردازیم. الگوریتم ارائه شده، به نوعی الگوریتمی حریصانه به شمار می‌رود.

برای مثال گراف روبرو را در نظر بگیرید.



ابتدا رئوس این گراف را با الگوریتمی که در ادامه گفته خواهد شد رنگ آمیزی می‌کنیم. که به صورت مقابل در خواهد آمد.

سپس یال‌ها را با توجه به الگوریتمی که به آن اشاره می‌کنیم رنگ می‌کنیم. در نتیجه گراف مقابل به دست می‌آید که گراف نهایی است.

این الگوریتم نیز محدودیت خاصی نداشته و بر روی هر نوع گرافی قابل اجراست.

به تعداد کمترین رنگ‌های لازم برای رنگ آمیزی کامل یک گراف، total chromatic number آن گراف می‌گویند. می‌توان اثبات کرد که این عدد همواره برای هر گراف حداکثر برابر با حاصل جمع بیشترین درجه‌ی گراف و یک عدد ثابت است. حدسی وجود دارد که بیان می‌کند این عدد ثابت برای مجموعه کوچکی از گراف‌ها مثل تمام گراف‌های دو بخشی، گراف‌های بدون طوقه و اکثر گراف‌های مسطح بجز آن‌هایی که دارای حداکثر درجه‌ی ۶ هستند، برابر با ۲ است. برای مثال‌های بالا این حدس درست بوده است. اگر این حدس درست باشد، می‌توان آن را به همه‌ی گراف‌های مسطح نیز تعمیم داد. این حدس به نام «حدس رنگ آمیزی کامل بهزاد-وایزینگ» مشهور است.

البته قابل ذکر است که هنوز هیچ الگوریتم کاملاً بهینه‌ای برای این مسئله معرفی نشده است. لذا بسته به الگوریتم مورد استفاده برای رنگ آمیزی گراف، ممکن است گاهی نتایج متفاوت باشند. اما غالباً نتیجه‌ای در محدوده‌ی حدس رنگ آمیزی کامل بهزاد-وایزینگ به دست می‌دهند.

در این الگوریتم فرضیاتی باید در نظر گرفته شوند که به شرح زیر هستند:

- گراف باید بدون جهت باشد.
- شماره رئوس اعدادی طبیعی باشند.
- در فایل ورودی، همواره بزرگترین شماره‌ی رأس، یکی کمتر از تعداد کل رئوس گراف است (چون شماره گذاری رئوس از صفر آغاز می‌شود). در ابتدای دریافت فایل ورودی، این عدد مشخص می‌شود.
- اگر عددی بود که در بازه‌ی صفر تا بزرگترین شماره رأس وجود داشت که در فایل ورودی نبود، به این معناست که رأس مذکور ایزوله است. این به این معناست که این الگوریتم توانایی اجرا بر روی گراف‌های ناهمبند را نیز دارد.

از نقاط ضعف این الگوریتم می‌توان به تعداد محدود رنگ‌ها (۱ تا ۱۱) اشاره کرد.

رنگ آمیزی رئوس گراف

برای رنگ آمیزی رئوس طبق الگوریتم زیر عمل می‌کنیم:

۱. اولین رأس را با اولین رنگ، رنگ آمیزی می‌کنیم.
۲. به رأس بعدی رفته و رنگ رئوس مجاورش را بررسی می‌کنیم.
۳. در بازه‌ی رنگ رئوس مجاورش، کمترین عددی که متعلق به هیچ یک از رئوس مجاور نبوده را به عنوان رنگ آن رأس در نظر می‌گیریم. اگر این شرایط موجود بود پس از انجام این مرحله به مرحله‌ی ۵ می‌رویم.
۴. اگر شرایط بالا برقرار نبود، یعنی تمام اعداد موجود در بازه‌ی اعداد رنگ‌های رئوس مجاور به حداقل یکی از رأس‌ها تعلق داشته باشد، اولین عدد خارج از این بازه را به عنوان رنگ آن رأس برمی‌گزینیم.
۵. تا وقتی همه رئوس رنگ آمیزی شوند مراحل ۲ تا ۵ را انجام می‌دهیم.
۶. پایان.

در پایان این الگوریتم تمامی رئوس رنگ شده اند.

پیچیدگی زمانی اجرای این الگوریتم به خاطر وجود دو حلقه‌ی for تودرتو، برابر با $O(|V|^2)$ است. توضیحات بیشتر داخل کامنت‌های کد آورده شده است.

رنگ آمیزی یال‌های گراف

برای رنگ آمیزی یال‌های گرافی که رئوس آن رنگ شده، از الگوریتم زیر استفاده می‌کنیم:

۱. اولین یال را انتخاب می‌کنیم.
۲. برای یال انتخاب شده رنگ دو رأس سازنده^۱ و یال‌های مجاورش را بررسی می‌کنیم.
۳. در بازه‌ی رنگ دو رأس سازنده و یال‌های مجاورش، کمترین عددی که متعلق به هیچ یک از آن‌ها نبوده را به عنوان رنگ آن یال در نظر می‌گیریم. اگر این شرایط موجود بود پس از انجام این مرحله به مرحله‌ی ۵ می‌رویم.

^۱ منظور از رئوس سازنده دو رأسی است که دو طرف یک یال قرار گرفته اند

۴. اگر شرایط بالا برقرار نبود، یعنی تمام اعداد موجود در بازه‌ی اعداد رنگ‌های رئوس سازنده و یال‌های مجاور به حداقل یکی از آن‌ها تعلق داشته باشد، اولین عدد خارج از این بازه را به عنوان رنگ آن یال انتخاب می‌کنیم.
۵. تا وقتی همه یال‌ها رنگ‌آمیزی شوند مراحل ۲ تا ۵ را انجام می‌دهیم.
۶. پایان.

در انتهای اجرای این الگوریتم، تمام یال‌ها نیز رنگ شده اند، به طوری که هیچ اشتراکی با یال‌ها یا رئوس مجاور ندارند. این الگوریتم ممکن است در بعضی نمونه‌ها همواره کمترین تعداد رنگ‌ها را به دست ندهد، چراکه برای کمترین تعداد رنگ‌ها هنوز الگوریتم کاملی ارائه و تأیید نشده است.

پیچیدگی زمانی اجرای این الگوریتم به دلیل وجود دو حلقه for تودرتو، برابر با حداکثر $O(|E||V|. (d + 2))$ است که در آن d بیشترین درجه‌ی رئوس گراف است. توضیحات بیشتر در کامنت‌های کد داده شده است.

ورودی و خروجی

ورودی برنامه یک فایل csv بوده که حاوی یال‌های گراف است. یال‌ها در این فایل با ذکر رأس‌های دو سرشان مشخص شده اند. در وهله‌ی اول، بزرگترین عدد را در فایل پیدا می‌کنیم تا درجه گراف مشخص شود. اگر عددی کوچکتر از بزرگترین عدد داخل فایل بود که در هیچ یالی نبود، به این معناست که آن رأس ایزوله است. این مسئله مشکلی در اجرای الگوریتم ایجاد نمی‌کند. ورودی‌ها خوانده و پردازش شده و تبدیل به گراف می‌شوند.

قابل ذکر است که ورودی‌های غیر مجاز عبارت اند از:

- اعداد غیرطبیعی
- حروف
- طوقه

که کاربر از استفاده از آن‌ها منع شده است.

خروجی این برنامه در یک فایل txt ذخیره می‌شود که شامل همه‌ی رئوس و یال‌ها به همراه رنگ‌شان است. همچنین عدد کروماتیک کامل این گراف نیز محاسبه می‌شود که از آنجا که در موارد خواسته شده در فایل خروجی نبود، تنها در نمایش گرافیکی گراف نمایش داده می‌شود.

ساختار پروژه

این پروژه از بخش‌های زیر تشکیل شده است:

۱. گراف

- ۱,۱. رأس: در این قسمت رأس‌هایی با ویژگی‌های رنگ، برچسب و مجموعه‌ی رئوس مجاور ساخته شده است.
- ۱,۲. یال: در این قسمت یال‌هایی با ویژگی‌های رنگ، رأس‌های سازنده و مجموعه‌ی یال‌های مجاور ساخته شده است.
- ۱,۳. گراف: در این قسمت با استفاده از رئوس و یال‌هایی که قبلاً ساخته شده، یک گراف بدون جهت می‌سازیم.

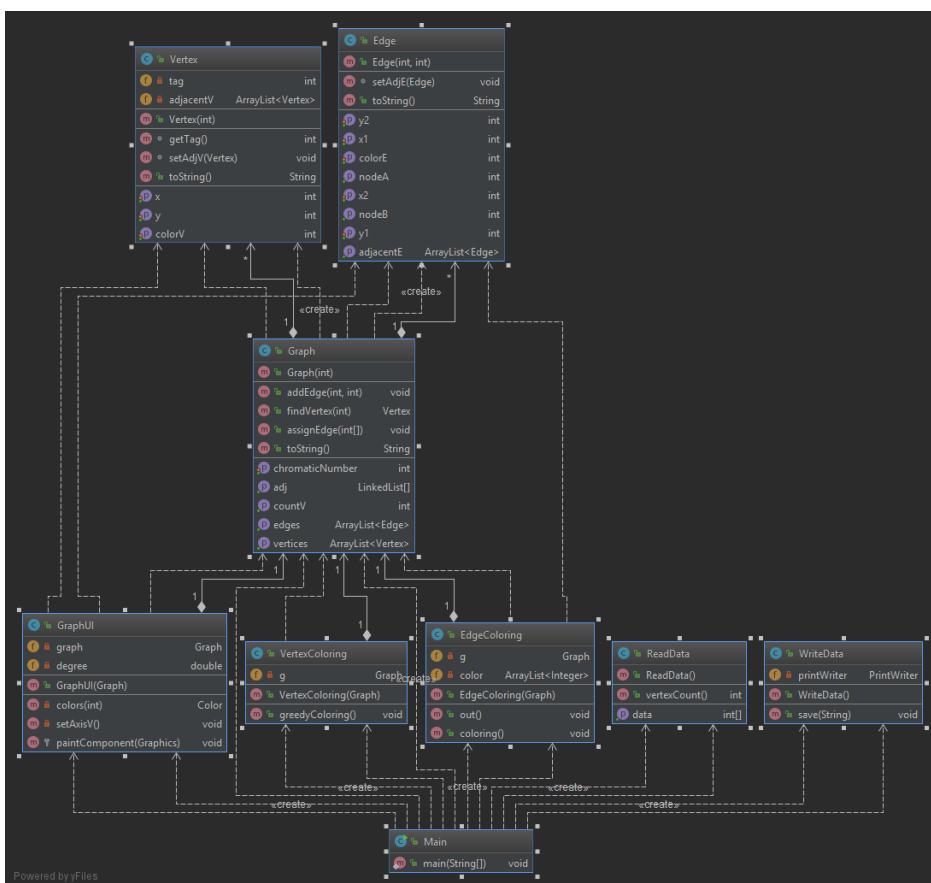
۲. رنگ آمیزی

- ۲,۱. رنگ آمیزی یال‌ها: در این قسمت به پیاده‌سازی الگوریتم گفته شده برای رنگ آمیزی یال‌ها پرداختیم.
- ۲,۲. رنگ آمیزی رئوس: در این قسمت به پیاده‌سازی الگوریتم گفته شده برای رنگ آمیزی رئوس پرداختیم.
۳. ورودی و خروجی: در این قسمت فایل CSV ورودی شامل یال‌های گرافی که قرار است ساخته شود را گرفته و یک فایل txt حاوی نتیجه نهایی اجرای برنامه تولید می‌کنیم.

۴. فایل‌ها: در این قسمت فایل‌های ورودی و خروجی و برخی فایل‌های مربوط به پروژه نگهداری می‌شود.

۵. نمایش گرافیکی: در این قسمت گراف نهایی پردازش شده و به صورت گرافیکی نمایش داده می‌شود.

۶. برخی جزئیات دیگر مانند کارهای هر تابع نیز در Javadoc تابع مربوطه ذکر شده است.



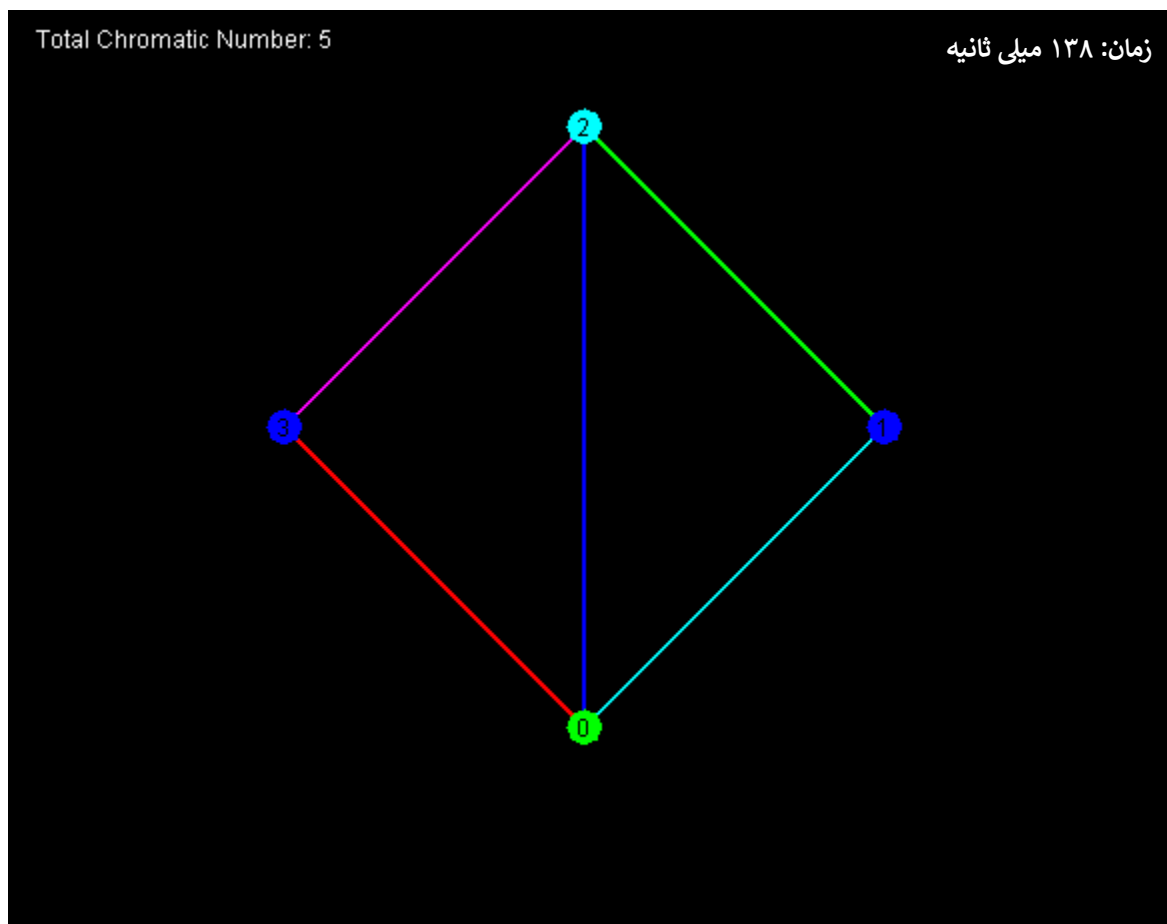
در روبرو نمودار شماتیک ساختار کلی برنامه قابل ملاحظه است:

نمونه‌ی اجراهای الگوریتم

سه نمونه از گراف‌های رنگ‌آمیزی شده توسط این الگوریتم در ذیل قابل مشاهده است.

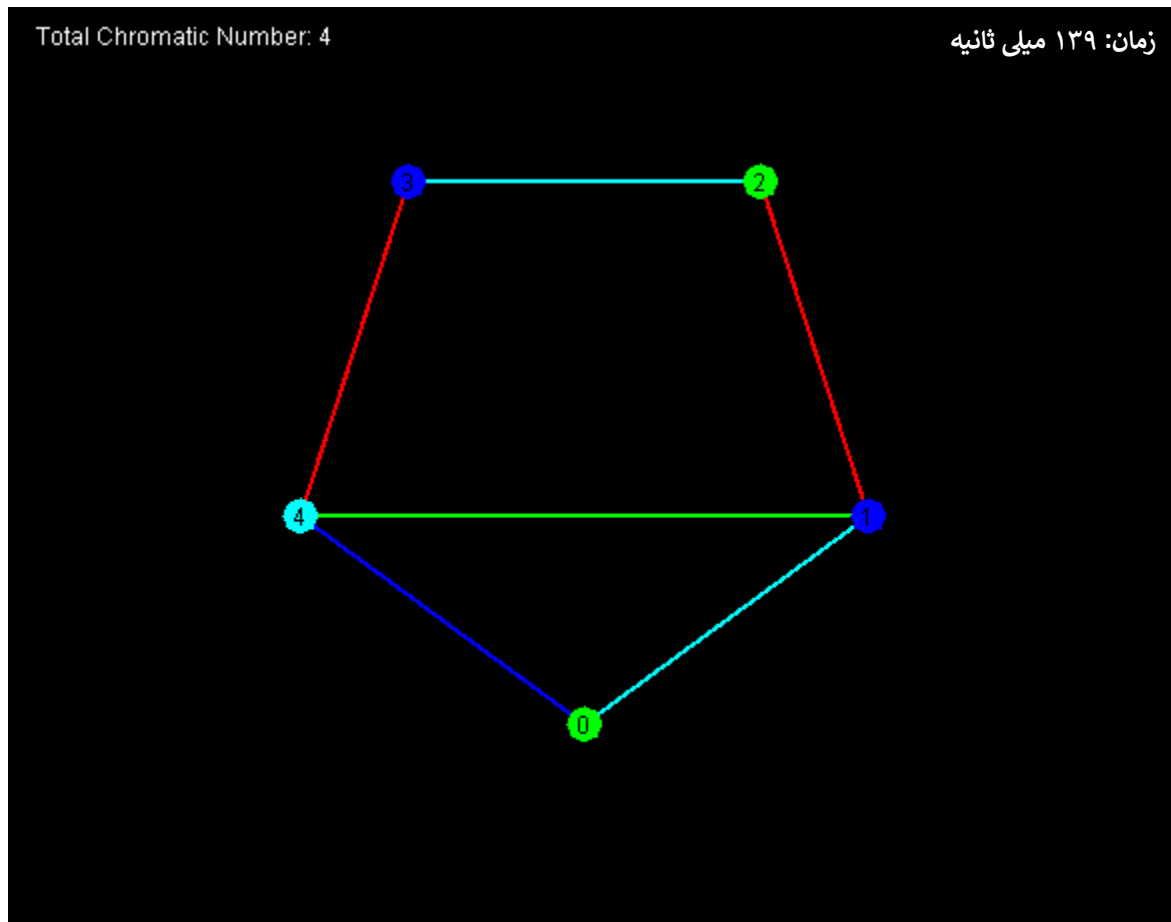
۱.

0, 1
0, 2
0, 3
1, 2
2, 3



۲.

0, 1
0, 4
1, 2
1, 4
2, 3
3, 4



۳.

0, 1
 0, 4
 0, 5
 0, 8
 1, 2
 1, 4
 1, 7
 1, 11
 2, 3
 2, 4
 2, 5
 3, 5
 5, 6
 5, 7
 7, 8
 7, 11
 7, 14
 7, 16
 8, 10
 8, 13
 8, 16
 10, 11
 15, 16

