باسمه تعالی

پروژه رنگ آمیزی گراف – رنگ آمیزی به روش backtracking - پرهام رحیمی – ۹۵۳۱۰۳۱

۱. کد اجرایی: همراه گزارش پیوست شده است.
۲. مقاله مورد نظر: از مقاله ای به طور مستقیم استفاده نشد تنها از سایت های مختلف برای یادگیری الگوریتم استفاده شد چرا که در تعریف پروژه ی داده شده لینکی برای تعریف پروژه این موضوع داده نشده بود.
۳. گزارش نهایی:

a. مدل Backtracking Algorithm به این صورت کار می کند: از راس اول شروع می کند و به آن یک رنگ نسبت می دهد سپس سراغ راس بعد رفته و از رنگ شماره یک شروع می کند به نسبت دادن رنگ ها به آن و هنگامی که به اولین رنگی رسید که همرنگ با هیچکدام از همسایه هایش نیست (امن است) آن رنگ را به آن نسبت می دهد و به سراغ راس بعد می رود ولی اگر هیچ رنگی را نمیتوانستیم به آن نسبت دهیم (در محدوده ی تعداد رنگ داده شده (نهایتا تعداد رئوس گراف)) به صورت Backtracking به راس های قبلی که آن را رنگ کرده باز می گردد (از آخر به اول) و رنگ آن را تغییر میدهد و باز میگردد. این کار را برای تک تک راس ها تکرار می کند.
در واقع Backtracking Algorithm نوعی Naive Algorithm است چرا که تمام حالات را چک می کند و یکی از حالاتی که درست است را خروجی میدهد.

b. با توجه به نوع الگوریتم، محدودیتی نداریم تنها ممکن است از نظر زمانی به زمان زیادی احتیاج داشته باشیم.

c. دقیقا مدل Backtracking Algorithm مشابه آنچه در بالا گفته شد پیاده سازی شده است با این تفاوت که یک ورودی m برای الگوریتم در نظر گرفته شده است که درواقع تعداد رنگ های موجود است این باعث می که بتوان از این الگوریتم استفاده های گسترده تری کرد ولی در حالت کلی m را برابر تعداد راس ها در نظر میگیریم تا در هر صورت جوابی به ما بدهد هر چند که جوابش "بیشینه انتشار" نباشد.
گراف ورودی از یک فایل csv. خوانده میشود و ماتریس مجاورت آن تشکیل می شود سپس رنگ های نسبت داده شده به همه ی راس ها ابتدا صفر در نظر گرفته می شود و از راس اول الگوریتم روی آن اجرا میشود تا جایی که همه ی راس ها رنگ شده باشند یا به این نتیجه رسیده باشیم که با تعداد رنگ های داده شده رنگ آمیزی ممکن نیست.

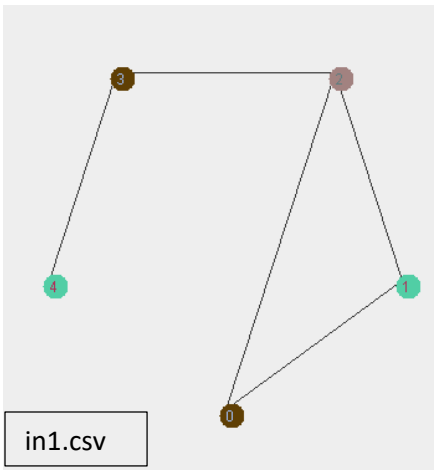
سپس ماتریس ورودی و رنگ های به دست آمده را به کلاس های گرافیکی می دهیم تا شکل گرافیکی آن را رسم کند.(امتیازی)

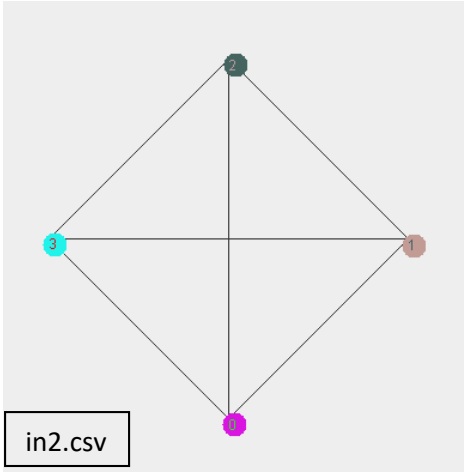
d. الگوریتم پیاده سازی شده کاملا مشابه مدل اصلی است و مانند آن در زمان آزاد هیچ محدودیتی ندارد.

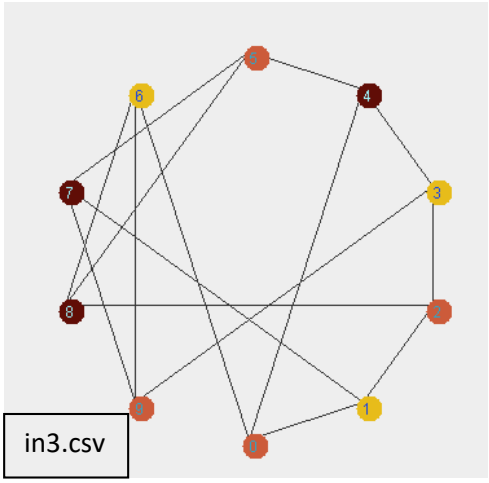
e. در پیاده سازی الگوریتم فرض شده که گراف به صورت csv. به آن داده می شود به طوری که در هر سطر عدد اول راس اول است و عدد دوم راس دوم که به آن یال دارد. و بدیهی است که فرض شده که تمام یال ها دوطرفه هستند.

f. از آنجا که تمام حالات بررسی می شود مرتبه زمانی الگوریتم بالا است. O(V^m) که در آن m تعداد رنگ هاست که ما برای حالت کلی برابر V در نظر گرفتیم پس میتوان گفت برای حالت کلی مرتبه زمانی الگوریتم ما O(V^V) است. چرا که در بدترین حالت تمام درخت تصمیم گیری حالات را برسی می کند.

g. سه ورودی in1.csv و in2.csv و in3.csv (که همراه پروژه پیوست شده اند) به آن ورودی داده شد و سه خروجی out1.txt و out2.txt و out3.txt (که همراه پروژه پیوست شده اند) متناظر با آن ها در زمان های(به ترتیب) ۲ میلی ثانیه، ۱ میلی ثانیه، ۲ میلی ثانیه خروجی داده شد.







ورودی گراف به صورت csv. به آن داده می شود به طوری که در هر سطر عدد اول راس اول است و عدد دوم راس دوم که به آن یال دارد. و خروجی یک فایل متنی است که در هر خط عدد اول شماره راس گراف و عدد دوم شماره رنگ نسبت داده شده به آن است.

تعداد راس ها و تعداد یال ها بر سرعت اجرای اگوریتم تاثیر دارند.

h. نقطه قوت الگوریتم کارا بودن آن برای تمام حالات گراف ورودی است.

نقطه ضعف آن زمان بسیار زیاد گرفته شده برای حل آن است.

i. با توجه به اینکه در الگوریتمی که پیاده سازی شده همانطور که توضیح داده شد یک ورودی m گرفته می شود که برا اساس آن امکان پذیری رنگ زدن با m رنگ خروجی داده می شود، می توان برای یک گراف با v راس، m را از ۱ شروع کرده و یکی یکی افزایش دهیم تا زمانی که خروجی true شود اولین m ای که به ازای آن خروجی true می شود "بیشینه انتشار" است.
از آنجا که برای هر m تمام حالات درخت حالات رنگ آمیزی با m رنگ بررسی میشود پس اولین m ای که خروجی true میگیریم قطعا کوچکترین m ممکن و کمترین تعداد رنگ ممکن برای رنگ آمیزی آن است.(امتیازی)

۴. همراه پروژه پیوست شده اند.