

## پروژه الگوریتم

در این پروژه برای نگه داری لیست یال های گراف از لیست دارای tuple استفاده میکنیم که هر tuple بیانگر یک یال بین دو راس هست.

(با توجه به تست کیس های داده شده فرض رو بر این گرفتیم که گراف رو بی جهت در نظر بگیریم تا راحت تر باشیم)

در ابتدای بایستی الگوریتم های مورد نیاز برای sort کردن آرایه در نظر گرفت که چهار الگوریتم Bubble Sort ,Quick ,Insertion ,Merge هستن پیاده سازی کرده ایم که با توجه به این که بایستی لیست ها رو بر حسب امتیازی که یال ها دارند بایستی sort شوند و کمترین امتیاز که یال ها دارند حذف شوند(زیرا کمترین وابستگی بین دو گراف نهایی هست نسبت به درجه رؤس یال هست)

به دو صورت میتوان از فایل یال ها را خواند یا از فایل txt یا از فایل های csv بخوانیم و در نهایت با توجه به حذف یال ها را تا زمانی جلو میبریم که به دو گراف غیر متصل تقسیم کنیم .

برای محاسبه Zij باید بین سه یال فرمتی برا قرار شود تا دور سه تایی که از رؤس i, j وجود دارد

برای بدست آوردن درجه راس ها باید فقط یک بار در لیست به دنبال تعداد یال ها بگردیم .

سپس برای هر یال ما میتوانیم مقدار امتیازی نسبت دهیم و با توجه به اینکه اگر یکی از رؤس مقدار 1 رو داشته باشد با توجه به فرمول تقسیم بر صفر خواهیم داشت مقدار انرا 1000000000 میگذاریم تا این یال در معرض حذف قرار نگیرد

برای اینکه زمان محاسبه هر صورت را بدست آوریم میتوانیم از time استفاده کنیم

در نهایت دو گراف نهایی رو که با حذف یال ها بدست آمده رو در یک فایل CSV ذخیره میکنیم

برای اینکه پروژه را تست کنیم دوتا تست کیس که در زمان پایینی جواب میدهد رو مورد بررسی قرار دادیم.

برای sort کردن بعضی اوقات یالهایی خواهیم داشت که امتیازشان یکی هست در این جا بدون در نظر گرفتن این روش رفتیم (به دلیل ضیق وقت) 😊

در این حالت بهتر است که در صورت تساوی امتیاز ها میتونیم به راس یال های نگاه کنیم و با توجه به کمینه بودن اونها اولویت حذف بدیم

برای اجرای پروژه در تابع `point_to_vertex` باید یک لیست و نوع سورت رو بدست بیاریم  
برای نوع لیست اگر `txt` باشه از `make_graph` و اگر `csv` باشد از `read_csv` کمک میگیریم

Book.csv و test13.txt (100 edge)