

مسیریاب

در این پروژه به دنبال یافتن کم ترافیک ترین مسیر بین دو شهر هستیم.

اما تنها چیزی که از نقشه شهرها می دانیم، تعداد ماشین های ورودی و خروجی هر معبر است.

با استفاده از جبر خطی می خواهیم ترافیک هر معبر را حساب کرده و بهترین مسیر را برای رسیدن از نقطه ای به نقطه دیگر پیشنهاد بدهیم.



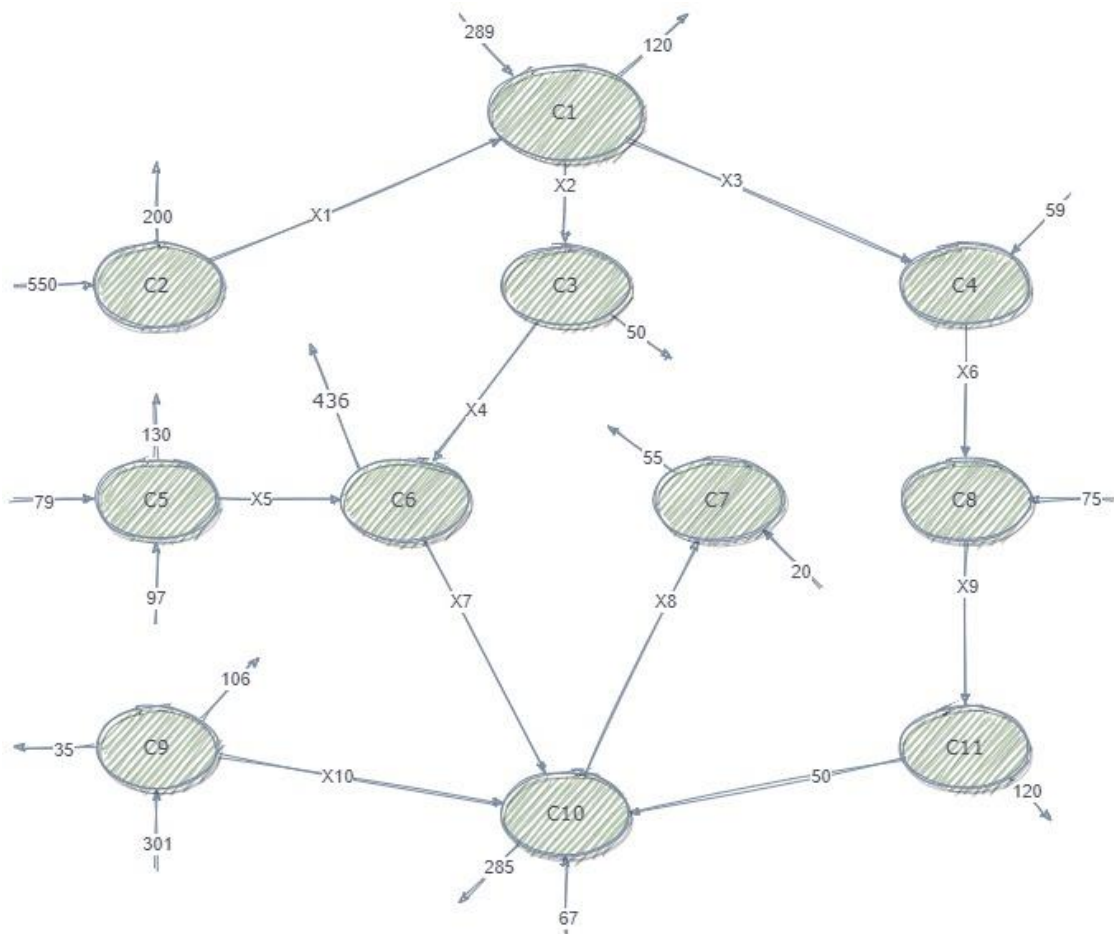
فهرست مطالب

.....	مسیریاب
2	تفسیر نقشه و تبدیل به ماتریس افزوده
5	حل معادله و پیدا کردن ترافیک هر معبر
5	توضیح الگوریتم کاهش سطری
7	پیدا کردن بهترین مسیر بین دو شهر (برای مطالعه‌ی بیشتر و کاربردها)
8	پیاده سازی
10	قوانین و ددلاین



تفسیر نقشه و تبدیل به ماتریس افزوده

در شکل زیر بخشی از نقشه راه‌های مواصلاتی بین شهرهای یک کشور رسم شده است.



هر شهر به وسیله تعدادی جاده به دیگر شهرها متصل است.

تعداد ماشین‌های ورودی و خروجی شهرها را داریم و به دنبال این هستیم که ترافیک بین شهری را محاسبه کنیم.

می‌دانیم که مجموع تعداد ماشین‌هایی که به شهر وارد می‌شوند با مجموع تعداد ماشین‌هایی که از شهر خارج می‌شوند، برابر است.

بنابراین می‌توانیم با برابر قرار دادن این دو عبارت، به یک معادله برای هر شهر برسیم.



حال وقتی به ازای هر شهر این معادله را بنویسیم، دستگاه معادلاتی تشکیل می‌شود که با حل آن، می‌توانیم ترافیک معابر بین شهری را بدست بیاوریم.

برای حل این دستگاه معادلات ما از جبر خطی و الگوریتم کاهش سطری¹ کمک می‌گیریم. ابتدا دستگاه معادلات را مشخص کرده و ماتریس افزوده² آن را می‌سازیم و سپس با کاهش سطری این ماتریس، به جواب دستگاه معادلات دست پیدا می‌کنیم. برای تبدیل نقشه به دستگاه معادلات، هر جریان ورودی را با ضریب +1 و جریان خروجی را با ضریب -1 مشخص می‌کنیم. برای نمونه جریان‌های ترافیک نقشه بالا را می‌توان با دستگاه معادلات زیر شبیه‌سازی کرد.

$$\left\{ \begin{array}{l} c_1: \quad x_1 - x_2 - x_3 + 289 - 120 = 0 \\ c_2: \quad \quad -x_1 + 550 - 200 = 0 \\ c_3: \quad \quad \quad x_2 - x_4 - 50 = 0 \\ c_4: \quad \quad \quad \quad x_3 - x_6 + 59 = 0 \\ c_5: \quad \quad -x_5 + 79 + 97 - 130 = 0 \\ c_6: \quad \quad \quad x_4 + x_5 - x_7 + 436 = 0 \\ c_7: \quad \quad \quad \quad x_8 + 20 - 55 = 0 \\ c_8: \quad \quad \quad \quad \quad x_6 - x_9 + 75 = 0 \\ c_9: \quad \quad -x_{10} + 301 - 35 - 106 = 0 \\ c_{10}: \quad x_7 - x_8 + x_{10} + 50 + 67 - 285 = 0 \\ c_{11}: \quad \quad \quad \quad \quad \quad x_9 - 50 - 120 = 0 \end{array} \right.$$

¹ Row reduction

² Augmented matrix



با ساده کردن هر یک از معادله های دستگاه معادلات بالا، به دستگاه معادله پایین می رسیم.

$$\left\{ \begin{array}{l} C_1: x_1 - x_2 - x_3 = -169 \\ C_2: \quad \quad -x_1 = -350 \\ C_3: \quad \quad x_2 - x_4 = 50 \\ C_4: \quad \quad x_3 - x_6 = -59 \\ C_5: \quad \quad -x_5 = -46 \\ C_6: x_4 + x_5 - x_7 = -436 \\ C_7: \quad \quad x_8 = 35 \\ C_8: \quad \quad x_6 - x_9 = -75 \\ C_9: \quad \quad -x_{10} = -160 \\ C_{10}: x_7 - x_8 + x_{10} = 168 \\ C_{11}: \quad \quad x_9 = 170 \end{array} \right.$$

حال که دستگاه معادلات ساده شده را داریم، می توانیم ماتریس افزوده این دستگاه را تشکیل دهیم.

$$\left[\begin{array}{cccccccccccc|c} 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -169 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -350 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 50 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -59 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -46 \\ 0 & 0 & 0 & 1 & 1 & 0 & -1 & 0 & 0 & 0 & -436 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 35 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & -75 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -160 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 1 & 168 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 170 \end{array} \right]$$



حل معادله و پیدا کردن ترافیک هر معبر

حال که ماتریس افزوده ترافیک را تشکیل دادیم، می‌توانیم با اعمال عملیات‌های ردیفی³، ماتریس افزوده را به فرم کاهش یافته نردبانی تبدیل کرده و جریان ترافیک هر معبر را بدست آوریم.

توضیح الگوریتم کاهش سطری

فرایند تبدیل یک ماتریس به فرم نردبانی کاهش یافته را می‌توان به دو مرحله تقسیم بندی کرد.

(1) تبدیل ماتریس به فرم نردبانی⁴

(2) تبدیل ماتریس از فرم نردبانی به فرم نردبانی کاهش یافته⁵

برای انجام مرحله اول، لازم است که در ابتدا، اولین عضو از سطر اول را چک کنید تا در صورتی که مقدارش برابر صفر باشد، این سطر را با یکی از سطرهایی که در ستون اول خود، دارای مقداری غیر صفر است، جابه‌جا⁶ کنید.

در ادامه باید تمامی عناصر ردیف اول را بر مقدار عنصر محوری در این ردیف تقسیم کنید (در این جا همان عنصر موجود در ستون اول) که در اصل از عملیات ردیفی تغییر مقیاس⁷ استفاده می‌کنیم. سپس باید با اعمال عمل جایگزینی ردیفی⁸ و جمع کردن ضریبی از ردیف اول در دیگر ردیف‌های ماتریس، مقدار تمامی عناصر زیر عنصر محوری در ستون اول را صفر کنید. با اعمال پی در پی این عمل بر روی ردیف و ستون دوم (دومین موقعیت محوری)، ردیف و ستون سوم (سومین موقعیت محوری) و ... در نهایت فرم نردبانی ماتریس اولیه تشکیل می‌شود.

³ Row operations

⁴ Forward substitution

⁵ Backward Substitution

⁶ Row Interchange

⁷ Row Scaling

⁸ Row Replacement



حال که مرحله اول انجام شد، به سراغ مرحله دوم می رویم.

برای تبدیل این ماتریس به فرم نردبانی کاهش یافته تنها نیاز است که از آخرین ردیف دارای موقعیت محوری شروع کنید.

از آنجایی که در مرحله Forward Substitution پس از رسیدن به هر ردیف، مقادیر آن ردیف را بر عنصر محوری آن ردیف تقسیم کرده ایم، بنابراین پس از پایان این مرحله، تمامی موقعیت های محوری برابر 1 خواهند بود.

پس با اعمال عملیات ردیفی جایگزینی بر روی ردیف های بالایی، تمامی عناصر بالای عنصر محوری در ردیف آخر را می توانید صفر کنید.

سپس این عمل را برای باقی عناصر محوری نیز انجام می دهیم.

در نهایت ماتریس حاصل، ماتریس به فرم نردبانی کاهش یافته خواهد بود.



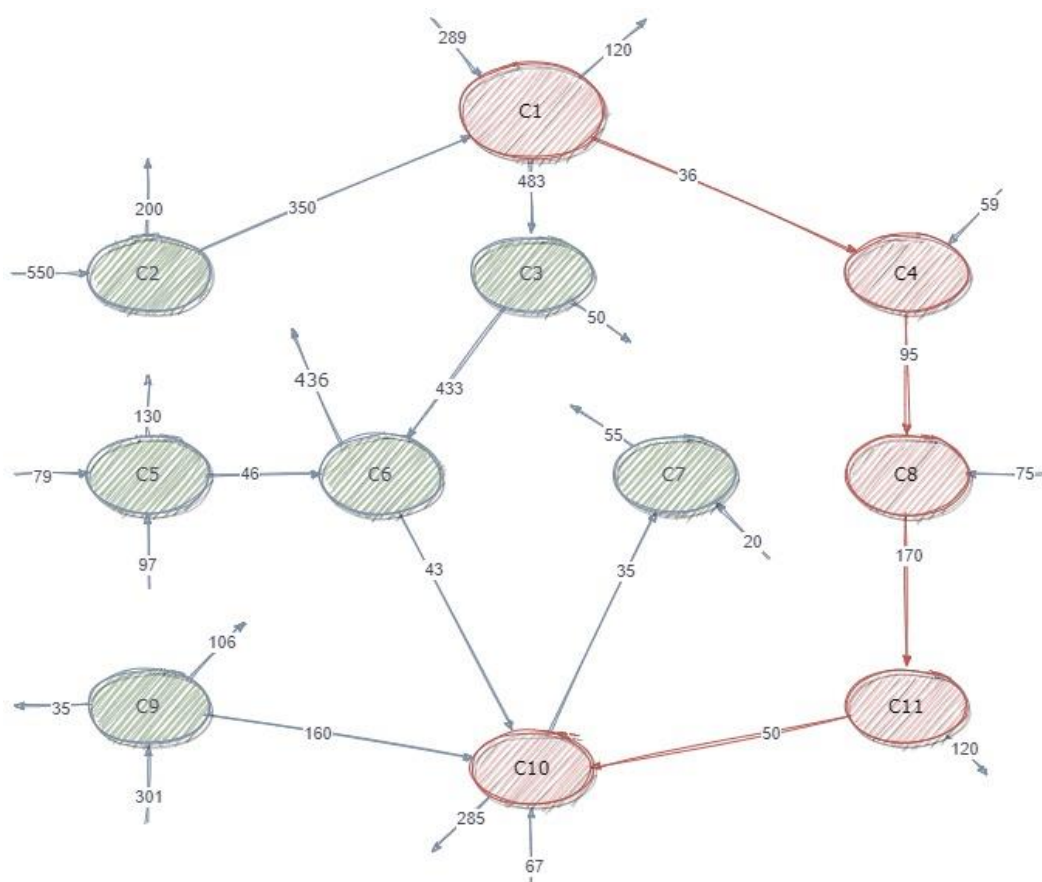
پیدا کردن بهترین مسیر بین دو شهر (برای مطالعه‌ی بیشتر و کاربردها)

در این پروژه فرض می‌کنیم که جاده‌ها طول یکسان دارند و ترافیک بین جاده‌ها، تنها عامل مهم در تعیین بهترین مسیر است.

حال که ترافیک نقشه مشخص شده است، می‌توانیم نقشه کشور را با یک گراف جهت دار وزن دار شبیه‌سازی کنیم.

هر شهر یک راس و هر جاده یک یال در این گراف است. وزن هر یال نیز با میزان ترافیک آن جاده مشخص می‌شود.

حال با استفاده از الگوریتم‌های یافتن کوتاه‌ترین مسیر در گراف (مثلاً الگوریتم Dijkstra)، می‌توانیم بهترین مسیر که در اینجا کم ترافیک ترین مسیر است را پیدا کنیم.
مثلاً شکل زیر بهترین مسیر برای حرکت از شهر $C1$ به $C10$ است.





پیاده سازی

برای این پروژه انتظار می‌رود کدی را پیاده‌سازی کنید تا با گرفتن ماتریس افزوده و حل دستگاه معادله مربوطه، بتوانیم به ترافیک هر معبر برسیم.

ورودی و خروجی کد شما باید به صورت زیر باشد:

ورودی:

در سطر اول دو عدد که به ترتیب از چپ به راست تعداد سطر ها (n) و تعداد ستون ها (m) می باشد، وارد می شود.

سپس در n سطر بعدی m عدد با فاصله از هم وارد می شود که در واقع ماتریس افزوده ی ورودی می باشد.

خروجی:

در خروجی ابتدا ماتریس کاهش یافته ی سطری نمایش داده می شود. و سپس در $m - 1$ خط بعدی متغیر ها به ترتیب و با فرمت $XN = P$ نمایش داده می شوند که XN نشان دهنده متغیر N ام و P نشان دهنده مقدار متغیر می باشد.

مثال ورودی 1:

```
3 4
1 0 0 2
0 0 1 6
0 1 0 5
```

مثال خروجی 1:

```
1 0 0 2
0 1 0 5
0 0 1 6
X1 = 2
X2 = 5
X3 = 6
```



مثال ورودی 2:

2 4
1 2 -1 4
2 -4 0 5

مثال خروجی 2:

1 0 -0.5 3.25
0 1 -0.25 0.375
 $X_1 = 8.25$
 $X_2 = 2.875$
 $X_3 = 10$

نکات:

- دقت شود که خواسته این پروژه صرفاً حل دستگاه معادله است.
- پیاده سازی شما باید برای حالت کلی حل دستگاه معادلات باشد و به ازای هر ابعادی از ماتریس افزوده باید جواب دستگاه معادلات را حساب کرده و جواب نهایی را برگرداند.
- در صورتی که پس از حل دستگاه به **متغیرهای آزاد** برخورد کردید، آنها را با مقدار ثابت ۱۰ جایگزین کنید.
- در صورت علاقه مند بودن به بخش کاربردی این پروژه یعنی یافتن بهترین مسیر، می توانید از توابع قرارداده شده در [اینجا](#) استفاده کنید.



قوانین و ددلاین

- هر دانشجو می‌بایست پروژه را به صورت انفرادی انجام دهد. تقلب‌ها به صورت خودکار، توسط سامانه کوئرا بررسی خواهد شد.
- از آن جایی که زبان برنامه نویسی پایتون، یکی از زبان های پرکاربرد در حوزه جبر خطی است و آموزش‌های مربوط به این زبان و کتابخانه‌های آن، توسط تیم تدریس‌یاری در اختیار شما قرار گرفته است، بنابراین برای پیاده سازی این پروژه تنها مجاز به استفاده از زبان پایتون و کتابخانه Numpy در کنار توابع و کتابخانه‌های پیش فرض پایتون هستید.
- استفاده از هر زبان برنامه نویسی یا کتابخانه ای دیگر قابل قبول نبوده و در صورت استفاده، نمره‌ای به شما تعلق نخواهد گرفت.
- از آنجایی که تصحیح به صورت خودکار توسط سامانه کوئرا انجام می‌شود، حتماً و حتماً فرمت ورودی و خروجی مربوط به پروژه را رعایت کنید.
- رعایت تمیزی کد، استفاده از توابع مختلف برای پیاده سازی پروژه به شدت استقبال می‌شود.
- ددلاین این پروژه تا ساعت **23:59 روز 3 آبان** خواهد بود و به هیچ عنوان تمدید نخواهد شد.

با آرزوی موفقیت و سلامتی

تیم تدریس یاری جبر خطی کاربردی، پاییز ۱۴۰۰