

تمرین ۱۰ درس طراحی الگوریتم

بنفشه کریمیان

سید صالح اعتمادی

دانشگاه علم و صنعت ۹۸-۹۷

لطفاً به نکات زیر توجه کنید:

- مهلت ارسال این تمرین ۴ خرداد ساعت ۱۱:۵۹ ب.ظ است.
 - این تمرین شامل سوال های برنامه نویسی می باشد، بنابراین توجه کنید که حتماً موارد خواسته شده را رعایت کنید. .
 - نام شاخه، پوشه و پول ریکوست همگی دقیقاً A10 باشد.
 - اگر در حل تمرین شماره‌ی ۱۰ مشکلی داشتید، لطفاً به [@BTS_BangTanARMY](#) مراجعه کنید.
- موفق باشید.

توضیحات کلی تمرین

تمرین این هفته ی شما، ۳ سوال دارد که باید به همه ی این سوال ها پاسخ دهید. برای حل این سری از تمرین ها مراحل زیر را انجام دهید:

۱. ابتدا مانند تمرین های قبل، یک پروژه به نام A10 بسازید.

۲. کلاس هر سوال را به پروژه ی خود اضافه کنید و در قسمت مربوطه کد خود را بنویسید. هر کلاس شامل دو متد اصلی است:

متد اول: تابع solve است که شما باید الگوریتم خود را برای حل سوال در این متد پیاده سازی کنید.

متد دوم: تابع process است که مانند تمرین های قبلی در TestCommon پیاده سازی شده است. بنابراین با خیال راحت سوال را حل کنید و نگران تابع process نباشید! زیرا تمامی پیاده سازی ها برای شما انجام شده است و نیازی نیست که شما کدی برای آن بنویسید.

۳. اگر برای حل سوالی نیاز به تابع های کمکی دارید؛ می توانید در کلاس مربوط به همان سوال تابع تان را اضافه کنید.

اکنون که پیاده سازی شما به پایان رسیده است، نوبت به تست برنامه می رسد. مراحل زیر را انجام دهید.

۱. یک UnitTest برای پروژه ی خود بسازید.

۲. فولدر TestData که در ضمیمه همین فایل قرار دارد را به پروژه ی تست خود اضافه کنید.

۳. فایل GradedTests.cs را به پروژه ی تستی که ساخته اید اضافه کنید. برای این تمرین مانند A8، A9 و A7 برای هر سوال تست جداگانه با زمان جداگانه در نظر گرفته شده. بعد از حل کردن هر تمرین Assert.Inconclusive را از ابتدای تست حذف کرده و آن را اجرا کنید. چنانچه سوالی را حل نکردید از Assert.Inconclusive استفاده کنید.

۴. دقت کنید که در این تمرین برخلاف تمرین های گذشته جواب نهایی موجود در فایل های تست با خروجی تابع Solve شما متفاوت است. یک Verifier برای این تمرین در TestCommon پیاده سازی شده که خروجی شما را به یک SatSolver میدهد و جواب آن را با جواب تست کیس مقایسه می کند. به همین منظور لازم است که بعد از merge کردن A10 در TestCommon.A10 و درست کردن و باز کردن Solution، روی پروژه TestCommon کلیک راست کرده و گزینه RestoreNugetPackages را انتخاب کنید تا بسته نرم افزاری Microsoft.Solver.Foundation نصب شود.

دقت کنید که TestCommon تغییر یافته است. بنابراین شما باید نسخه ی جدید آن را با دستور Pull دریافت کنید .

۱ قراردادن فرکانس برای برج های شبکه جی اس ام

در این مسئله شما یاد میگیرید که مسئله ی دادن فرکانس به شبکه gsm را به مسئله ی رنگ کردن یک گراف با ۳ رنگ کاهش دهید. سپس الگوریتمی طراحی میکنید که این مسئله را به یک SAT تبدیل میکند. شبکه های gsm برای برقراری ارتباط بین تلفن های همراه اند که شامل فرستادن اطلاعات؟ در محیطی که با فرکانس های مختلف کار میکند میشود. در هر شش ضلعی به طور معمول یک برج شبکه قرار میگیرد که به آن cell می گویند. (به همین خاطر به تلفن همراه phone cell میگویند) هر تلفن همراه از بین برج های موجود برجی با بیشترین قدرت را انتخاب میکند. برای این منظور برج های نزدیک و در همسایگی یکدیگر باید از فرکانس های مختلف استفاده کنند. شما باید روی مسئله ی دادن فرکانس به برج های مختلف کار میکنید و ۳ فرکانس مختلف در دست دارید. همسایگی ها برای شما مشخص است و شما باید امکان دادن فرکانس های مختلف برای این شبکه با ۳ فرکانس را بررسی کنید. در واقع این مسئله با مسئله ی معروف رنگ آمیزی گراف برابر است و شما یک گراف به عنوان ورودی دریافت میکنید. رنگ ها همان فرکانس ها، نودها همان برج ها و یال ها همان همسایگی برج ها باهم اند. این مسئله یک مسئله ی complete np است و راه حل بهینه ای برای حل آن را نمیدانیم پس شما باید آنرا به یک SAT کاهش دهید که اغلب با برنامه هایی به اسم SAT-solver ها قابل حل اند.

فرمت ورودی: خط اول ورودی دو عدد n یا تعداد نودها و m یا تعداد یال ها را دارد. شماره گذاری نودها از ۱ تا n است و m خط بعدی شماره ی نودهایی که بهم متصل اند رو بصورت دو به دو دارد. (هر یال نمیتواند به خودش متصل شود) فرمت خروجی: شما باید یک فرمول بولین را به فرم CNF با فرمتی که معرفی میکنیم به خروجی دهید. اگر مسئله قابل حل است و جوابی وجود دارد که هر نود گراف یک رنگ داشته باشد و هر دو یال کنار هم یک رنگ نباشند، فرمول شما باید قابل حل باشد در غیر اینصورت فرمول شما باید غیر قابل حل باشد. تعداد متغیرهای فرمول شما نباید بیشتر از ۳۰۰۰ و تعداد نامساوی ها نباید بیشتر از ۵۰۰۰ باشد. در خط اول خروجی دو عدد c تعداد نامساوی ها و v تعداد متغیرها را قرار دهید. برای متغیرهای خود از ۱ تا v یک ایندکس قرار دهید. دقت کنید که متغیرها به فرم x_1 یا x_2 بصورت x_1 or x_2 است. در c خط بعدی میبایست ابتدا ایندکس متغیرهای فرمول را قرار دهید سپس یک ۰ در انتها قرار دهید که بتوان پایان خط را تشخیص داد. برای مثال آورده شده باید ۱ ۲ ۳ ۷ ۰ در خروجی قرار گیرند. not هر متغیر را با یک منفی قبل از ایندکس ان نشان دهید. دقت داشته باشید که هر عدد بجز عدد اخر باید عددی غیر صفر بین -v تا v باشد و $1 < C < 5000$ است. $1 \leq V \leq 3000$ اگر تعداد فرمول هایی که میتوانند امکان پذیر بودن مسئله را نشان دهند بیشتر از ۱ بودند هر کدام از آنها را میتوانید به خروجی دهید خروجی ها با یک SAT-solver چک میشوند.

راهنمایی: برای هر نود سه متغیر بولین قرار دهید که نشان دهنده ی هر رنگ باشند و در صورت یک شدن آن رنگ به آن نود داده میشود. حال باید به گونه ای ضوابطی مانند هر نود تنها ی رنگ باید داشته باشد و نودهای

همسایه میبایست رنگ های متفاوت داشته باشند را با این متغیرها به فرم CNF بنویسید.
 مثال ۱) در این مثال دقت کنید که گراف ورودی قابل رنگ آمیزی با ۳ رنگ به طوری که همسایه ها رنگ های متفاوتی داشته باشند است پس هدف این است که در خروجی یک فرمول بدهید که زمانی که به یک SAT Solver داده میشود حل شود و جوابی داشته باشد. مهم این است که برای گراف های ممکن خروجی در SAT Solver حل شود و برای ناممکن ها حل نشود و فرمول های ممکن یکتا نیستند. خروجی فرمول زیر x یا $\text{not } x$ است که نمونه ی ساده ای از فرمول های قابل حل است.

3 3
1 2
2 3
1 3
1 1
1 -1 0

مثال ۲) مثال زیر غیر قابل حل است پس فرمولی غیرقابل حل باید به خروجی بدهیم که x و $\text{not } x$ به خروجی داده شده است که نمونه ی ساده ای از فرمول های غیرقابل حل است.

4 6
1 2
1 3
1 4
2 3
2 4
3 4
2 1
1 0
-1 0

۲ تمیز کردن آپارتمان

در این مسئله شما بررسی میکنید که آیا امکان پذیر است که یک آپارتمان را بعد از طک مهمانی به گونه ای تمیز کنید که اثری از مهمانی در آن باقی نماند. شما باید این مسئله را به مسئله ی مسیر همیلتونی تبدیل و یک

الگوریتم برای تبدیل آن به SAT پیاده کنید. شما یک مهمانی در خانه داشتید و میخواهید خانه را تمیز کنید به طوری که اثری از مهمانی باقی نماند. برای این منظور شما بعد از تمیز کردن یک اتاق نمیتوانید دوباره از آن عبور کنید به دلیل اینکه ممکن است اثری از مهمانی باقی بگذارید. پس شما باید از یک اتاق به اتاق دیگری که در همسایگی اتاق قبلی است بروید، همه ی اتاق ها را تمیز کنید و از هر اتاق فقط یک بار عبور کنید. این مسئله میتواند به مسئله ی معروف مسیر همیلتونی تبدیل شود، یعنی با دادن یک گراف امکان یافتن راهی که از همه ی نود ها فقط یکبار میگذرد بیابید. اتاق ها همان نود ها و همسایگی آن ها همان یال ها اند. مسیر همیلتونی یک مسئله ی complete np است و راه حل بهینه ای برای آن نداریم. شما میتوانید با کاهش آن به یک SAT به کمک یک SAT-solver آن را بهینه حل کنید.

فرمت ورودی: خط اول ورودی دو عدد n یا تعداد اتاق ها و m یا تعداد اتاق های متصل بهم را دارد. شماره گذاری اتاق ها از ۱ تا n است و m خط بعدی شماره ی اتاق هایی که بهم متصل اند رو بصورت دو به دو دارد. درها دو طرفه اند یعنی از هر دو اتاق در همسایگی یکدیگر میتوان به دیگری رفت. دیهی است که هر در هر اتاق را به خودش وصل نمیکند. دقت کنید که هر در ممکن است چندبار به ورودی داده شود و یا در بین دو اتاق u و v بصورت $u v$ یا $v u$ نمایش داده شود. فرمت خروجی: شما باید یک فرمول بولین را به فرم CNF با فرمتی که معرفی میکنیم به خروجی دهید. اگر مسئله قابل حل است و بتوان از یک یال شروع کرد و از هر یال حتما یکبار بدون گذشتن از یال های تکراری گذشت، فرمول شما باید قابل حل باشد در غیر اینصورت فرمول شما باید غیرقابل حل باشد. دقت کنید که تعداد متغیرها باید از ۱۲۰۰۰۰ تا کمتر باشد. در خط اول خروجی دو عدد c تعداد نامساوی ها و v تعداد متغیرها را قرار دهید. برای متغیرهای خود از ۱ تا v یک ایندکس قرار دهید. دقت کنید که متغیرها به فرم x_1 یا x_2 بصورت x_1 or x_2 فرمت هر خط فرمول بصورت x_1 or x_2 است. در c خط بعدی میبایست ابتدا ایندکس متغیرهای فرمول را قرار دهید سپس یک ۰ در انتها قرار دهید که بتوان پایان خط را تشخیص داد. برای مثال آورده شده باید ۱ ۲ ۳ ۷ ۰ در خروجی قرار گیرند. not هر متغیر را با یک منفی قبل از ایندکس آن نشان دهید. دقت داشته باشید که هر عدد بجز عدد اخر باید عددی غیر صفر بین $-v$ تا v باشد. اگر تعداد فرمول هایی که میتوانند امکان پذیر بودن مسئله را نشان دهند بیشتر از ۱ بودند هرکدام از آنها را میتوانید به خروجی دهید خروجی ها با یک SAT-solver چک میشوند. راهنمایی: به ازای هر نود یا اتاق به تعداد اتاق ها متغیر تعریف کنید که این متغیرها نشان دهنده ی ترتیب آنها در مسیر باشد. برای مثال اگر ۳ اتاق داشته باشیم و برای اتاق اول داشته باشیم ۱۰. یعنی این اتاق دومین اتاقی است که آن را تمیز میکنیم. حال باید ضوابطی نظیر: تمیز کردن حتمی هر اتاق، خالی نبودن یک ترتیب در مسیر (برای مثال دومین اتاقی که تمیز میکنیم باید وجود داشته باشد)، هر اتاق تنها یکبار در مسیر باشد، هیچ دو اتاقی در یک ترتیب مسیر قرار نگیرند (برای مثال اتاق اول در مسیر تنها یک اتاق میتواند باشد نه چندتا) و همسایگی (داشتن در) دو اتاق متوالی در مسیر را به فرم CNF بنویسید.

مثال ۱)

5 4
1 2
2 3
3 5
5 4
1 1
1 -1 0

مثال (۲)

4 3
1 2
1 3
1 4
2 1
1 0
-1 0

۳ جاگذاری بودجه ی تبلیغات

در تمرین قبلی شما روی تبلیغات آنلاین کار کردید. در این مسئله شما برای یک شرکت بزرگ که برای پیشرفت از تبلیغات استفاده میکند کار میکنید. شما باید تعیین کنید که آیا امکان پذیر است که بودجه ی تبلیغات را به همراه تمام ضوابط رعایت کنید یا خیر. شما یاد میگیرید که این مسئله را به یک $\text{programming Linear Integer}$ تبدیل کنید و سپس الگوریتمی طراحی میکنید که این مسئله را به SAT کاهش دهد. شرکت شما زیرمجموعه های زیادی دارد که روی تبلیغات تلوزیونی، رادیویی، اینترنتی و غیره کار میکنند. همه ی آن ها برنامه های تبلیغاتی خود را ریخته اند و شما برای پوشش همه ی این برنامه ها بودجه ندارید. شما باید برنامه ی بودجه را فردا تحویل دهید و زمان کافی برای مطالعه ی پیشنهاد همه ی زیرمجموعه ها ندارید. شما تصمیم میگیرید که یک پیشنهاد را به طور کامل رد و یا قبول کنید ولی با ضوابط بسیاری روبه رو میشوید. برای مثال بودجه ی کل تبلیغات شما محدود است و یک سری قرارداد ها با بعضی شرکت های تبلیغاتی دارید که شما را مجبور به صرف یک میزان هزینه ی حداقلی برای آن نوع تبلیغات میکند و صرف نظر از آن جریمه سنگینی دارد. ضوابط دیگری نیز همانند اجبار برای گذاشتن حداقل ۱۰ درصد بودجه یا یک ملیون در ماه برای تبلیغات تلوزیونی به منظور به یاد موندن برند شما برای مردم نیز میتوانند تعریف شوند. تمام این ضوابط و شروط را میتوان به فرم $\text{integer programming linear}$ باز نویسی کرد. X_i میتواند قبول شدن یا نشدن پیشنهاد i در نظر گرفت و شرایط را باز نویسی کرد. برای مثال قابل پوشش دادن هزینه تمام پیشنهاد های پذیرفته شده با بودجه ای که داریم را میتوان به صورت ... نوشت. شما ضوابط را به صورت $\text{programming linear integer}$ دریافت میکنید و میبایست آن را به SAT کاهش دهید. دقت کنید که در این مسئله حداکثر ۳ متغیر با کوئفیشنت غیر ۰ در هر نامساوی وجود دارد.

فرمت ورودی: در خط ورودی دو عدد n تعداد نامساوی ها و m تعداد متغیرها را دریافت میکنید. در n خط بعدی توضیحات ماتریس کوئفیشنت نامساوی ها، A که $n \times m$ بعدی است را دریافت میکنید. هر خط m عدد صحیح دارد که حداکثر ۳ عدد از آنها غیر ۰ است. خط آخر n عدد صحیح که وکتور b را تشکیل میدهند را دارد. برای سیستم نامساوی $Ax \leq b$ باید بررسی کنید که آیا وکتور x ای وجود دارد که این سیستم را پوشش دهد یا خیر. فرمت خروجی: شما باید یک فرمول بولین را به فرم CNF با فرمتی که معرفی میکنیم به خروجی دهید. اگر مسئله قابل حل است و جوابی برای آن وجود دارد که تمام ضوابط را پوشش میدهد، فرمول شما باید قابل حل باشد در غیر اینصورت فرمول شما باید غیر قابل حل باشد. تعداد متغیرهای فرمول شما نباید بیشتر از ۳۰۰۰ و تعداد نامساوی ها نباید بیشتر از ۵۰۰۰ باشد. در خط اول خروجی دو عدد c تعداد نامساوی ها و v تعداد متغیرها را قرار دهید. برای متغیرهای خود از ۱ تا v یک ایندکس قرار دهید. دقت کنید که متغیرها به فرم باینری اند. فرمت هر خط فرمول بصورت $x_1 \text{ or } x_2 \text{ or } x_3 \text{ or } x_7$ است. در c خط بعدی میبایست ابتدا ایندکس متغیرهای فرمول را قرار دهید سپس یک ۰ در انتها قرار دهید که بتوان پایان خط را تشخیص داد. برای مثال آورده شده باید $1 \ 2 \ 3 \ 7$ در خروجی قرار گیرند. not هر متغیر را با یک

منفی قبل از ایندکس ان نشان دهید. دقت داشته باشید که هر عدد بجز عدد اخر باید عددی غیر صفر بین $-v$ تا v باشد و $1 < C < 5000$ است. $1 \leq v \leq 3000$ اگر تعداد فرمول هایی که میتوانند امکان پذیر بودن مسئله را نشان دهند بیشتر از ۱ بودند هرکدام از انها را میتوانید به خروجی دهید خروجی ها با یک SAT-solver چک میشوند.

(مثال ۱)

2 3
5 2 3
-1 -1 -1
6 -2
1 1
-1 1 0

(مثال ۲)

3 3
1 0 0
0 1 0
0 0 1
1 1 1
1 1
1 -1 0

(مثال ۳)

2 1
1
-1
0 -1
2 1
1 0
-1 0
2 3
1 1 0
0 -1 -1
0 -2

2 1
1 0
-1 0