

پروژه ارضای محدودیت

توضیحات اولیه

این پروژه، مربوط به حل یک پازل است. در این پازل، به شما یک جدول داده میشود. هنگام شروع، تعدادی دایره سفید و سیاه در این جدول قرار داده شده اند و بقیه خانه های جدول خالی هستند. هدف، پر کردن این جدول است به صورتی که در انتها تمامی خانه های جدول پر شده باشند و محدودیت ها به صورت کامل ارضا شده باشند.

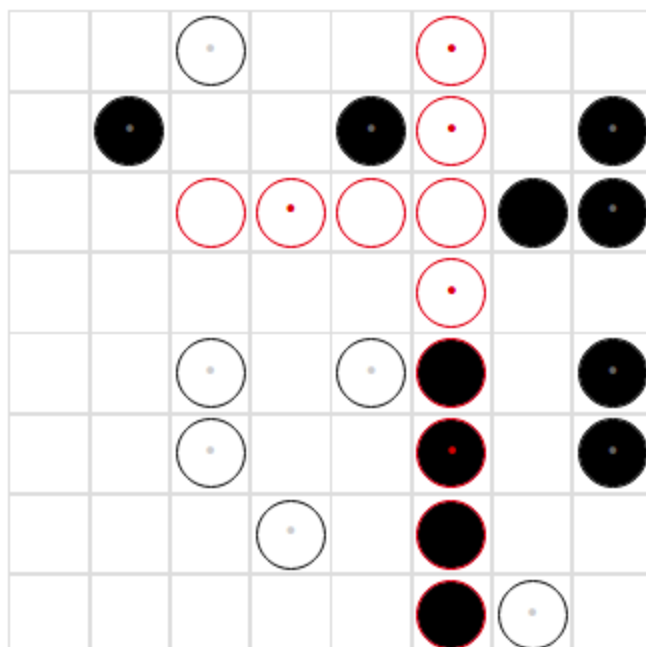
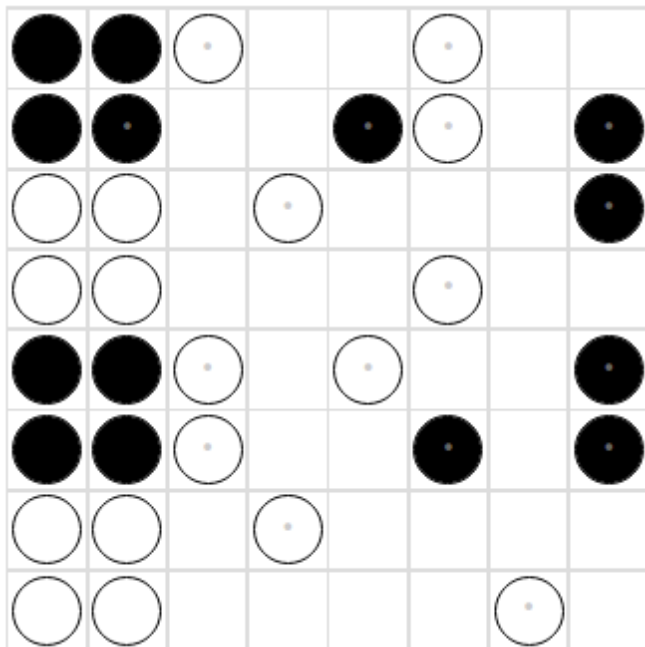
محدودیت ها به شرح زیر می باشند:

۱- هر سطر و ستون جدول باید شامل تعدادی برابر از دایره های سیاه و سفید باشد. برای مثال اگر جدولمان 8×8 باشد، باید در هر سطر و ستون چهار دایره سفید و چهار دایره سیاه وجود داشته باشد. (در حالت کلی اگر جدولی $n \times n$ داشته باشیم، باید در هر سطر و ستون، $\frac{n}{2}$ دایره سیاه و $\frac{n}{2}$ دایره سفید وجود داشته باشد.)

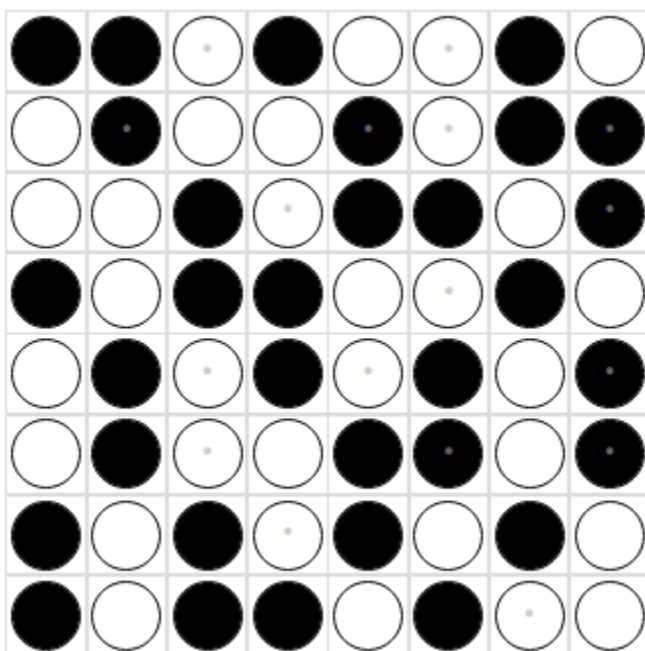
۲- بیش از ۲ دایره مشابه نمی توانند در یک سطر یا یک ستون، با یکدیگر مجاور باشند. (برای مثال، سه دایره سفید و سه دایره سیاه نمی توانند در یک سطر یا یک ستون، با یکدیگر مجاور باشند.)

۳- سطرها و ستون ها باید یکتا باشند و نباید یک سطر یا ستون تکراری به وجود بیاید.

برای درک بهتر، به مثال های صفحه بعد توجه کنید:



شکل ۱ و ۲ - حالت های غیر مجاز: در شکل سمت راست، بیش از ۲ دایره هم شکل کنار هم قرار گرفته اند و در شکل سمت چپ، دو ستون هم شکل داریم.



شکل ۳ - حالت مجاز (دایره هایی که درونشان نقطه دارند، دایره های پیش فرض هستند).

پیاده سازی

ابتدا باید مسئله را به صورت یک مسئله ارضای محدودیت مدل سازی کنیم.

1. متغیرها شامل خانه هایی از جدول هستند که در ابتدا به صورت پیش فرض دایره ای در آنها قرار داده نشده است. برای مثال اگر جدولی $n \times n$ داشته باشیم و m دایره پیش فرض قرار داده شده باشند،

$$n^2 - m$$

متغیر خواهیم داشت.

2. دامنه شامل دایره سیاه یا دایره سفید برای هر متغیر است.

3. محدودیت ها عبارت از مواردی هستند که در صفحات قبل توضیح داده شد.

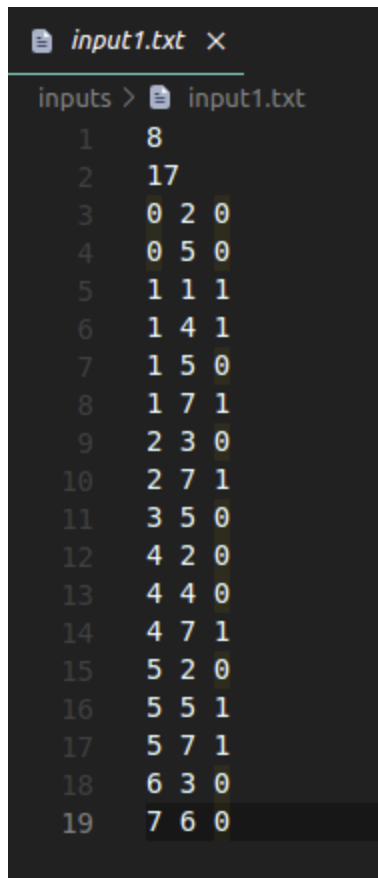
سپس باید با استفاده از الگوریتم Backtracking دارای Forward Checking، و توابع ابتکاری MRV و LCV، مسئله را حل کنیم.

قسمت مدل سازی توسط تیم حل تمرین پیاده سازی شده و شما باید الگوریتم های مربوط به حل را پیاده سازی کنید.

در انتها الگوریتم AC3 را پیاده سازی کرده و تاثیر استفاده از آن را بررسی کنید.

ورودی

در سطر اول فایل ورودی، عدد n که نشان دهنده اندازه طول جدول میباشد وجود دارد و در سطر دوم، عدد m که نشان دهنده تعداد دایره های پیش فرض است وجود خواهد داشت. در m سطر بعد، به ترتیب سطر، ستون و رنگ هر دایره پیش فرض مشخص شده است. توجه داشته باشید که در فایل ورودی 0 به معنی دایره سفید و 1 به معنی دایره سیاه میباشد و indexing در فایل ورودی از 0 میباشد.



The image shows a screenshot of a file editor window titled 'input1.txt'. The window displays a list of 19 inputs, each consisting of a number followed by a space and then a set of numbers. The inputs are numbered 1 through 19 on the left. The data for each input is as follows:

| Input | Data |
|-------|-------|
| 1 | 8 |
| 2 | 17 |
| 3 | 0 2 0 |
| 4 | 0 5 0 |
| 5 | 1 1 1 |
| 6 | 1 4 1 |
| 7 | 1 5 0 |
| 8 | 1 7 1 |
| 9 | 2 3 0 |
| 10 | 2 7 1 |
| 11 | 3 5 0 |
| 12 | 4 2 0 |
| 13 | 4 4 0 |
| 14 | 4 7 1 |
| 15 | 5 2 0 |
| 16 | 5 5 1 |
| 17 | 5 7 1 |
| 18 | 6 3 0 |
| 19 | 7 6 0 |

شکل ۴ - نمونه ورودی برای
مثال توضیح داده شده در بالا

خروجی

در خروجی، حالت اولیه مربوط به ورودی و حالت نهایی حل شده را به همراه زمان سپری شده برای حل مسئله نمایش دهید. (متدی برای نمایش حالت ها مطابق شکل موجود در صفحه بعد طراحی شده است.)

```
Initial Board:
-- -- ■ -- -- ■ -- --
-- ■ -- -- ■ ■ -- -- ■
-- -- -- ■ -- -- -- ■
-- -- -- -- -- ■ -- --
-- -- ■ -- -- ■ -- -- ■
-- -- ■ -- -- ■ -- -- ■
-- -- -- ■ -- -- --
-- -- -- -- -- -- ■
-- -- -- -- -- --

Result Board:
● ● ■ ● ○ ■ ● ●
○ ■ ● ● ■ ■ ● ■
○ ● ● ■ ● ● ● ■
● ● ● ● ● ■ ● ●
○ ● ■ ● ■ ● ● ■
○ ● ■ ● ● ■ ● ■
● ● ● ■ ● ● ● ●
● ● ● ● ● ● ■ ●

Total time: 0.109500137
arya@arya:~/Documents/artificial-
```

شکل ۵ - نمونه ای برای خروجی:
دایره های پیش فرض با مربع نمایش
داده شده اند.

نکات تکمیلی

1. این پروژه باید به صورت انفرادی پیاده سازی شود.
2. استفاده تدریجی از Git در روند پیاده سازی پروژه، دارای نمره اضافه خواهد بود.

موفق باشید