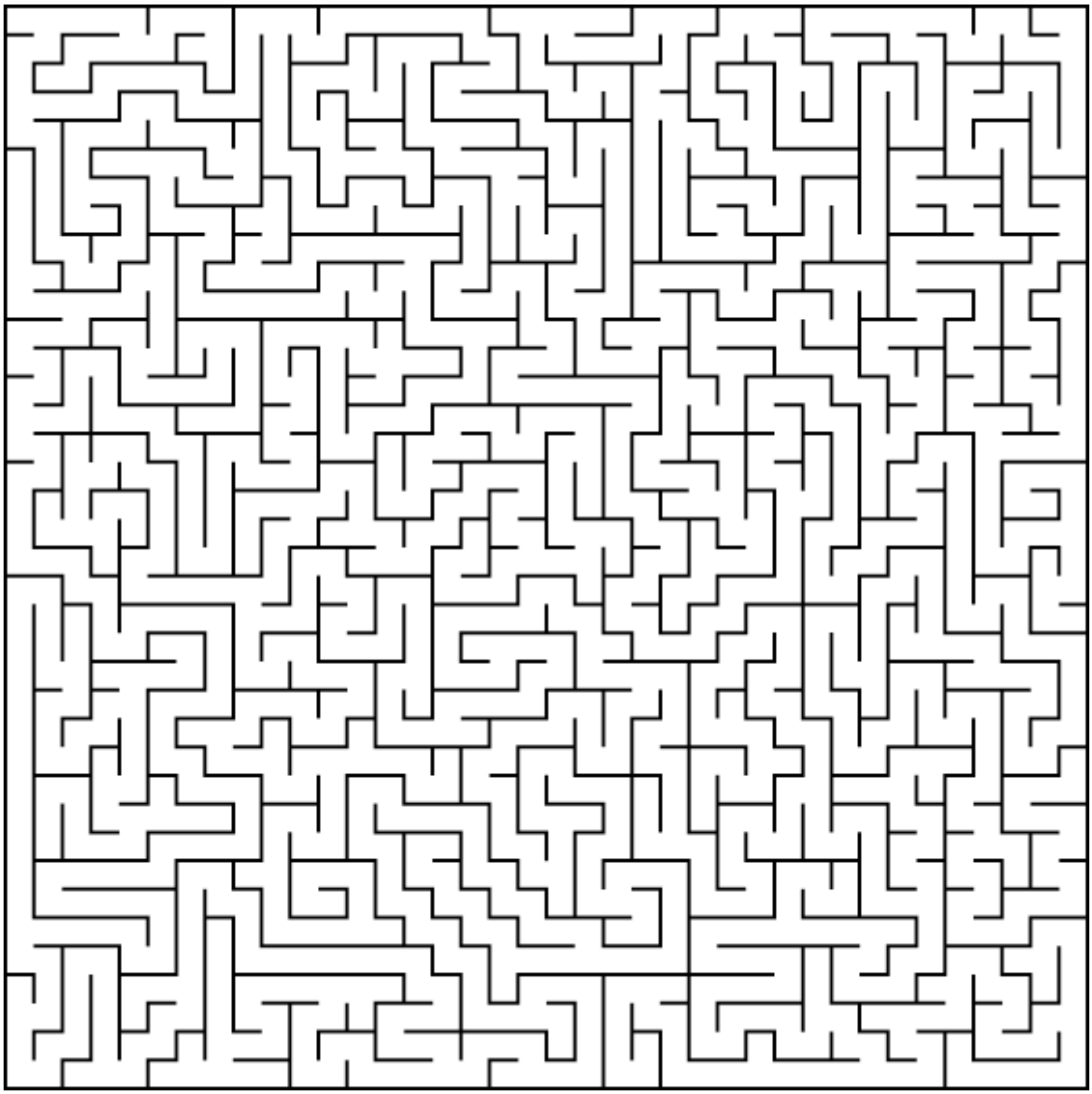
اینجا آخر راهه!

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: 10 مگابایت



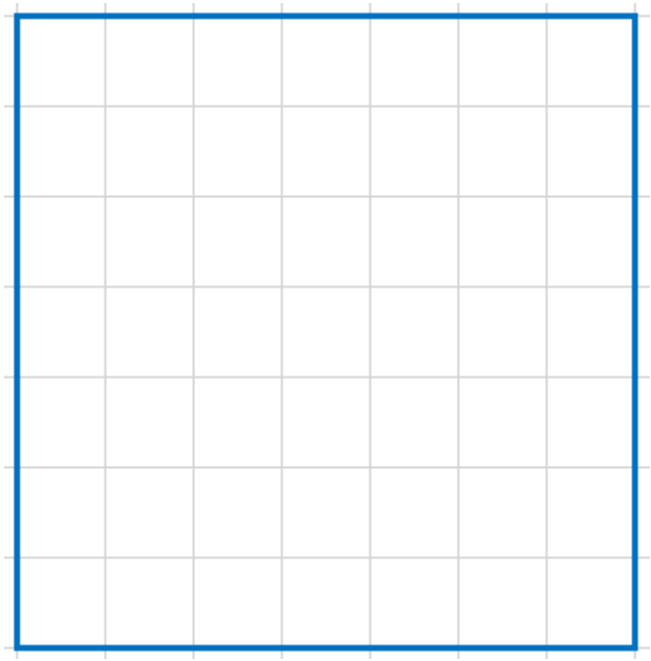
دانشجویان درس ساختمان داده که کم کم به پاپان راه این درس رسیده اند در می یابند که برای پایان این درس باید از مازی (قدیمی :))) به طول m (تعداد ردیف های جدول ماز) و عرض n (تعداد ستون های جدول

ماز) عبور کنند.در همین حین به خاطر می آورند که خوشبختانه برنامه نویسی بلد هستند پس به سرعت دست به کد می‌شوند تا در کم ترین زمان نقشه احتمالی این ماز را بیابند و این درس را نیز با موفقیت پشت سر بگذارند.

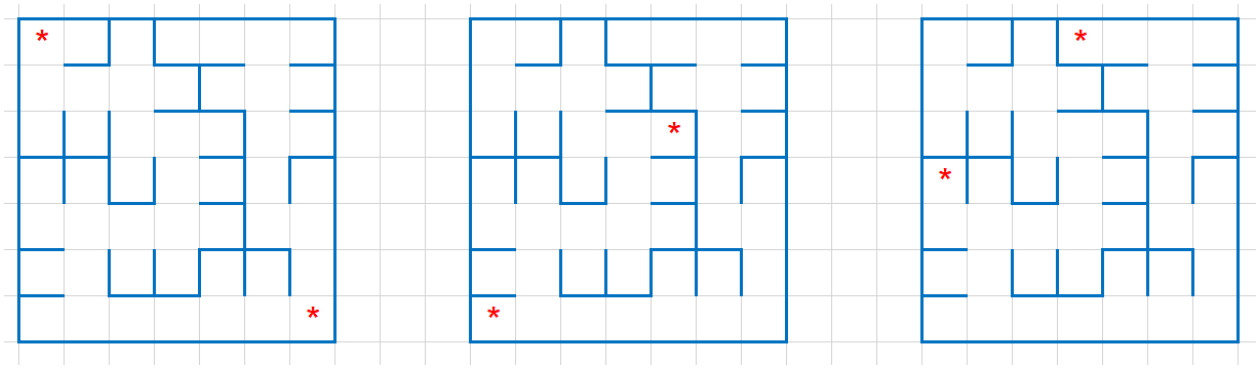
آنان می دانند که در هر روز توانایی t بار بررسی کل ماز را دارند پس باید برنامه ای بنویسد که با گرفتن m (طول ماز) و n (عرض ماز)، t حالت و نقشه متفاوت رندوم از ماز را ترسیم کند تا آنان بتوانند با آن نقشه ها به بررسی ماز بپردازند. نکته ای که باید به آن دقت کنید این است که آنان اصلا فرصت برای بررسی نقشه های تکراری را ندارند و باید هر t نقشه خروجی برنامه شما حداقل در یک خانه تفاوت داشته باشند!

این ماز ویژگی های زیر را باید داشته باشد:

- تمامی دیوارهای داخل ماز، حداقل به یک دیوار متصل هستند و دیواری وجود ندارد که به هیچ دیواری متصل نباشد. هم‌چنین تمامی دیوارها به صورت مستقیم یا غیرمستقیم به دیواره اطراف ماز وصل هستند.
- بین هر دو خانه مربعی ماز، تنها یک مسیر یکتا وجود دارد.



دیواره ی دور ماز در تصویر بالا قابل مشاهده است.



در تصویر بالا مشاهده می شود که بین هر دو خانه در ماز دقیقا یک مسیر یکتا وجود دارد.

راهنمایی TA های مهربان

شما باید این نقشه ماز را با استفاده از یک گراف مدل سازی کنید و سپس با الگوریتم های گرافی مرتبط با خواسته مسئله، ماز مربوط به مسیر را پیدا کنید.

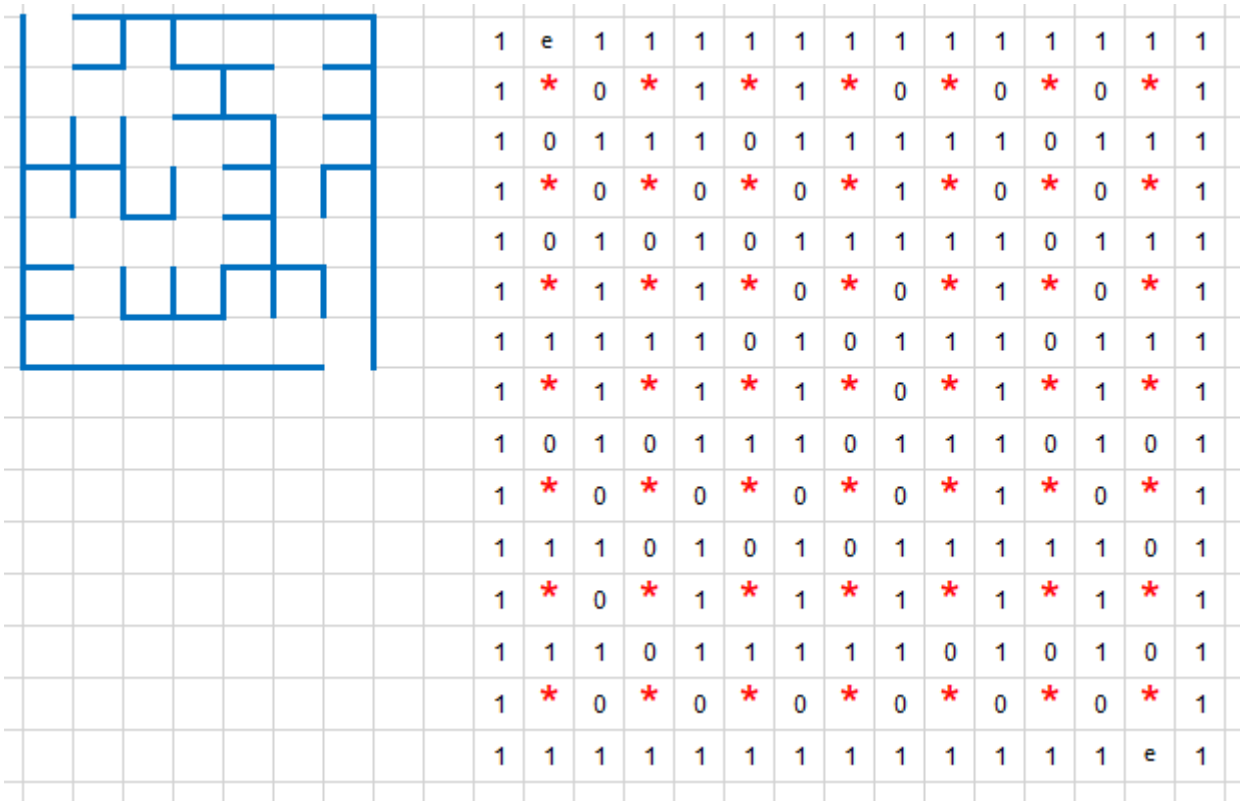
ورودی

ورودی به ترتیب شامل سه عدد m و n و t است. m عرض ماز و n طول آن است. همچنین t تعداد مازهای متفاوتی است که باید در خروجی چاپ شود.

$$1 \leq n, m, t \leq 100$$

خروجی

خروجی به تعداد t جدول به عرض $2 * m + 1$ و طول $2 * n + 1$خواهد بود که هریک به فرمت زیر است. تمامی خانه های ماز (نقاط قرمز رنگ نشان داده شده) می بایست با * در خروجی نشان داده شوند. تمام دیواره های اطراف هر خانه از جدول ماز بدین صورت نشان داده می شوند که در صورت وجود دیوار آن را با `1` و در غیر این صورت (در صورتی که دیواری بین دو خانه وجود نداشت و امکان عبور وجود داشت) آن را با `0` نشان می دهیم.همچنین برای ماز یک ورودی و یک خروجی در نظر گرفته شده است که جای ثابتی دارند که در شکل نیز مشخص است. این ورودی و خروجی ماز با کاراکتر `e` نمایش داده میشوند. در تصویر زیر یک ماز 7*7 و نحوه خروجی دادن آن مشخص شده است:



مثال

ورودی نمونه ۱

```
3 6 3
```

خروجی نمونه ۱

```
1e11111111111
1*0*0*0*0*0*1
1110111111101
1*1*0*1*0*1*1
1011111010101
1*0*0*0*1*0*1
11111111111e1

1e11111111111
```

```
1*1*0*0*0*0*1
1011111110101
1*0*0*0*0*1*1
1111111111101
1*0*0*0*0*0*1
11111111111e1

1e11111111111
1*0*0*1*0*1*1
1011101010101
1*0*1*0*1*0*1
1110111111101
1*0*1*0*0*0*1
11111111111e1
```

توجه داشته باشید که با توجه به اینکه ماز ها به طور رندوم ساخته میشوند، این تنها یک خروجی درست برای این ورودی است.

ورودی نمونه ۲

```
4 11 3
```

خروجی نمونه ۲

```
1e11111111111111111111
1*0*1*0*0*1*1*0*0*1*0*1
11101011101010101011101
1*0*1*1*1*1*0*1*1*0*1*1
10111010101111101110101
1*1*0*1*1*0*1*0*1*0*1*1
10111010111010111011101
1*0*0*1*0*0*0*1*0*0*0*1
11111111111111111111e1

1e11111111111111111111
1*0*0*0*0*0*0*0*0*0*0*1
```

```
111110111110111111111101
1*0*1*1*0*0*1*0*0*1*0*1
101011101111101011101111
1*1*0*1*0*1*0*1*1*0*1*1
111110111010111010111101
1*0*0*0*0*1*0*1*0*0*0*1
11111111111111111111e1

1e11111111111111111111
1*0*1*0*1*0*0*1*0*0*0*1
101010101110101011111101
1*1*0*1*1*0*1*0*1*0*1*1
111111101011111111010101
1*0*1*0*1*1*0*0*0*1*0*1
101010111010111111111111
1*1*0*0*0*1*0*0*0*0*0*1
11111111111111111111e1
```

زنجیره بلوک

در این سوال میخوایم یک بلاک چین پیاده سازی کنیم.

برنامه اصلی دارای یک کلاس Blockchain است که در واقع یک لینک لیست یک طرفه از بلاک های سیستم است که باید در هر زنجیره بلاک ها (blockchain) همواره نود انتهایی ذخیره شود.(tail)

این کلاس تابع های زیر را دارد :

- کانستراکتور که همراه آن بلاک اول نیز ساخته میشود.
- Create_block

در این تابع یک بلاک ساخته شده و به بلاک های قبلی اضافه میشود.

- Get_previous_block

در این تابع میتوان به بلاک قبلی دسترسی داشت.

- Proof_of_work

در این تابع اثبات کار بررسی و محاسبه میشود. به این صورت که اعداد مختلفی چک و بررسی میشود تا بتوان Nonce مناسب برای یک بلاک جدید را یافت. در اینجا Nonce باید به گونه ای یافت شود که اگر به عنوان ورودی به تابع F داده شد، چهار کاراکتر اول خروجی آن به صورت 0000 باشد.

F = Hash (Nonce ^ 2 - Previous_Nonce ^ 2)

- Hash

این تابع هش کل بلاک داده شده به آن را محاسبه میکند. در واقع به این صورت عمل میکند:

hash = Hash(index + Timestamp + Nonce + previous_hash)

(در صورت استرینگ بودن داده ها، اسکی کد آنها در فرمول قرار میگیرد.)

در این تابع میتوانید از هش های آماده مانند md5 , sha256, ... استفاده کنید.

امتیازی : یک هش با فرمول دلخواه خود، پیاده سازی کنید و آن را توضیح دهید.(در یک فایل پی دی اف)

هر بلاک دارای اطلاعات زیر است:

Index :

که نشان دهنده اندیس آن بلاک در بلاک چین اصلی است.

Timestamp :

که نشان دهنده زمان (روز، ماه،سال، ساعت، دقیقه) ساخته شدن آن بلاک است.(با فرمت دلخواه)

Nonce :

نانس مربوط به هر بلاک در آن قرار میگیرد.

Previous_hash :

مقدار هش بلاک قبلی در آن قرار دارد (برای اولین بلاک میتواند 0 باشد.) هر بلاک یک هش دارد که در اینجا به این صورت بدست می آید.

hash = Hash(index + Timestamp + Nonce + previous_hash)

پ ن :این هش با خروجی تابع F متفاوت است و هر کدام کاربرد متفاوتی دارند.

موارد امتیازی :

- ایجاد یک تابع Mine_block که بتوان با استفاده از آن بلاک های جدیدی ماین کرد و به بلاک چین اضافه نمود و همچنین برای کار با آن یک رابط کاربر تحت خط فرمان نوشت که بتوان هر بار دستورات زیر را به آن داد و خروجی را مشاهده کرد.

MineNewBlock :

یک بلاک جدید استخراج کند و به زنجیره اضافه کند و آن را نمایش دهد.

ShowChain :

کل بلاک ها را با اطلاعات کامل نمایش دهد.

Show x :

در صورت وجود، اطلاعات بلاک با ایندکس x را نمایش دهد.

توضیح مختصر : هر بلاک در بلاک چین دارای یک Nonce منحصر به فرد است. تنها زمانی یک بلاک جدید به زنجیره بلاک ها اضافه میشود که بتوان یک نانس جدید یافت. این عدد یک ویژگی خاصی دارد که یافتن آن را دشوار میکند. مثلا در اینجا باید Nonce یافت شده به گونه ای باشد که اگر به عنوان ورودی به تابع F داده شد، چهار کاراکتر اول آن 0000 شود.

F = Hash (Nonce ^ 2 - Previous_Nonce ^ 2)

برای آشنایی بیشتر میتوانید به لینک زیر مراجعه کنید.

https://www.youtube.com/watch?v=SSo_EIwHSd4&ab_channel=SimplyExplained

بیشتر و بهتر بدانیم :

https://www.youtube.com/watch?v=bBC-nXj3Ng4