

## مسئله ۸ وزیر

در این مسئله با فرض ورودی

```
[1, 0, 0, 0, 0, 0, 0, 0]
[0, 1, 0, 0, 0, 1, 0, 0]
[0, 0, 0, 0, 0, 0, 1, 0]
[0, 0, 1, 0, 1, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 1]
[0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 1, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0]
```

به مل مسئله با ۴ شیوه مختلف از راه مل تپه نوردی پرداخته ایم.

### 1. استاندارد: در این راه مل فروبی ها به شرح زیر بود:

تعداد نود دیده شده: ۱۶۸

تعداد نود باز شده: ۲

مسیر:  $[ \{ 5' = 7, 1' = 5 \} ]$

ارزش مقدار آخر: ۱

### 2. اول بهترین: در این راه مل فروبی ها به شرح زیر بود:

تعداد نود دیده شده: ۲۲۴

تعداد نود باز شده: ۳

مسیر:  $[ \{ 5' = 0, 7' = 14, 0' = 5 \} ]$

ارزش مقدار آخر: ۱

### 3. تصادفی: در این راه مل فروبی ها به شرح زیر بود:

تعداد نود دیده شده: ۱۶۸

تعداد نود باز شده: ۲

مسیر:  $[ \{ 5' = 7, 1' = 5 \} ]$

ارزش مقدار آخر: ۱

### 4. شروع تصادفی: در این راه مل فروبی ها به شرح زیر بود:

تعداد نود دیده شده: ۱۶۸۰

تعداد نود باز شده: ۲۱

مسیر:  $[ \{ 4' = 2, 0' = 3 \} ]$

ارزش مقدار آخر: ۰

البته لازم به ذکر هست که در شروع تصادفی مقدار اولیه تغییر کرده است. همچنین در انتخاب تصادفی و شروع تصادفی هم فروبی های مختلفی وجود داشت که فقط یکی از آنها را نمایش داده ایم. در بین راه مل ها روش استاندارد ممکن هست بفاطر مناسب نبودن حالت اولیه در یک مینیمم یا یک شانه گیر کند که همانطور که مشاهده میشود در این شانه گیر کرده است. این برای اولین بهترین هم صادق هست.

در جستجوی تصادفی نیز امکان به مینیمم سراسر نرسیدن وجود دارد. شاید بفاطر انتخاب اشتباه به جواب غیر مینیمم سراسری برسد.

در جستجوی شروع تصادفی متما به جواب مینیمم سراسری میرسیم ولی با صرف هزینه زمان.

## مسئله گراف

در این مسئله ورودی اولیه

۰ ۱ ۱ ۰ ۰ ۱ ۰  
۱ ۰ ۱ ۱ ۰ ۰ ۰  
۱ ۱ ۰ ۰ ۱ ۰ ۱  
۰ ۱ ۰ ۰ ۱ ۱ ۰  
۰ ۰ ۱ ۱ ۰ ۰ ۰  
۱ ۰ ۰ ۱ ۰ ۰ ۱  
۰ ۰ ۱ ۰ ۰ ۱ ۰

میباشد و طبق آن مسئله حل شده است.

در این مسئله ۳ روش مختلف برای سرد کردن در نظر گرفته شده است. بعلت طولانی بودن فروجی اصلی فقط تعداد نود های دیده شده و بسط داده شده و مقدار آخر را نمایش داده شده است. عدد -۱ به معنای گسسته شدن یال بین این دو نود هست.

1. **خطی:** بصورت خطی هر دفعه از ۱۰۰ عدد ۵ را کم میکنیم. یعنی طی ۲۰ بار اجرای ملقه به جواب نهایی میرسیم.

فروجی به شرح زیر است:

تعداد نود های دیده شده: ۹۰

تعداد نود های بسط داده شده: ۱۹

مقدار نهایی:

[۰, -۱, -۱, ۰, ۰, ۰, ۰, ۱, ۰]

[۰, -۱, ۰, ۱, ۱, ۰, ۰, ۰, ۰]

[۰, -۱, ۱, ۰, ۰, ۰, ۱, ۰, -۱]

[۰, ۰, ۱, ۰, ۰, ۰, ۰, -۱, ۰]

[۰, ۰, ۰, ۱, ۱, ۰, ۰, ۰, ۰]

[۱, ۰, ۰, ۰, -۱, ۰, ۰, ۰, ۱]

[۰, ۰, ۰, -۱, ۰, ۰, ۰, ۰, ۰]

ارزش مقدار نهایی: ۵

2. **لگاریتمی:** در هر دفعه عدد ۱۰۰ را بر ۲ تقسیم میکنیم. این ملقه به تعداد خیلی زیادی تکرار خواهد شد(فقط با

پایتون ۳ تعداد آن زیاد خواهد بود)

فروجی به شرح زیر است:

تعداد نود های دیده شده: ۵۳۷۱

تعداد نود های بسط داده شده: ۱۰۸۱

مقدار نهایی:

[۰, ۰, ۱, ۱, ۰, ۰, ۰, -۱, ۰]

[۱, ۰, ۰, ۱, -۱, ۰, ۰, ۰, ۰]

[۱, ۰, ۰, ۰, ۰, -۱, ۰, ۰, -۱]

[۰, ۰, -۱, ۰, ۰, ۰, ۱, ۱, ۰]

[۰, ۰, ۰, ۱, ۱, ۰, ۰, ۰, ۰]

[۰, -۱, ۰, ۰, ۱, ۰, ۰, ۰, ۱]

[۰, ۰, ۰, -۱, ۰, ۰, ۰, ۱, ۰]

ارزش مقدار نهایی: ۵

3. **مذری:** هر دفعه جذر مقدار احتمال را میگیریم تا به ۱ برسد. اینجا عدد را از ۱۰۰۰۰ شروع کردیم.

فروجی به شرح زیر است:

تعداد نود های دیده شده: ۱۸

تعداد نود های بسط داده شده: ۳

مقدار نهایی:

[۰, ۰, ۱, -۱, ۰, ۰, ۰, ۱, ۰]

[۱, ۰, ۰, -۱, ۱, ۰, ۰, ۰, ۰]

[۰, -۱, ۰, ۰, ۰, ۰, ۱, -۱, ۱]

[۰, ۰, ۱, ۰, ۰, ۰, ۱, ۱, ۰]

[۰, ۰, ۰, ۱, ۱, ۰, ۰, ۰]  
[۱, ۰, ۰, ۱, ۰, ۰, ۰, ۱]  
[۰, ۰, ۱, ۰, ۰, ۰, ۱, ۰]  
ارزش مقدار نهایی: ۷

همانطور که میدانیم از الگوریتم با اجراهای مختلف فروجی های مختلفی خواهند داشت ولی طبق آزمایش ها در راه مل  
۲ که تعداد اجراهای ملقه زیاد بود جواب های بهتری پیدا میشد. ولی این احتمال در روش های ۱ و ۳ که تعداد کمتری  
اجرا میشدند بوضوح کمتر بود. هرچه سرعت کاهش کمتر باشد و تعداد آن زیاد باشد شانس پیدا کردن جواب بهتر بیشتر  
است.

## مسئله معادله

در این مسئله چند حالت تصادفی ساخته و طبق فایل ضمیمه شده کد اجرا میشود.

الف) عکس های این ۳ حالت مختلف طی مرور زمان برای یک فروبی در فایل equation قرار گرفته است.

ب) اگر احتمال وفادار جهش را زیاد کنیم احتمال رسیدن به جواب بهینه نیز بیشتر میشود. اما این مشروط به آن هست که جهش را اعداد با فوبی انجام دهیم، در غیر اینصورت نسل های بدتری تولید خواهند شد. برای مثال اگر ما عدد جهش را تا ۴۰ بگذاریم نسل رفته رفته با هر جهش بدتر میشود ولی اگر آن را ۱۰ بگذاریم به مراتب رشد بهتر خواهد شد.

ج) بنظر میرسد که در تعداد کم فیلو الگوریتم خوب عمل نمیکند و در تعداد اجراهای کمی به جواب بهینه میرسند. همچنین اگر عدد را فیلو هم بالا ببریم همین نتیجه حاصل میشود. اما وقتی در تعداد میانه یعنی ۳۰۰ تا ۵۰۰ قرار میدهیم بهترین جواب ها حاصل میشود.

د) فیلو فیلو سریع تر به جواب خواهیم رسید. برای مثال اگر تعداد را ۵ بگذاریم تقریباً با ۳۰ بار تکرار ملقه به جواب میرسیم. اما وقتی تعداد را ۱۰۰ میگذاریم اکثراً زیر ۱۰ به جواب خواهد رسید.