

دوباره گیر افتاد

• نوع تمرین: گروهی

سفینه حالا در بُعد پنجم به یک شبکه‌ی عجیب با تونل‌های چندقطبی رسیده که علاوه بر انرژی مثبت و **منفی**، برخی تونل‌ها انرژی مغناطیسی خاصی دارند که اگر چندبار پشت‌سرهم وارد آن‌ها شوی، باعث تخلیه ناگهانی سیستم می‌شود.

🚧 قوانین مسیر:

۱. مسیر باید از $(0,0)$ به $(n-1,n-1)$ بره.
۲. فقط حرکت به راست و پایین مجازه.
۳. مجموع کل انرژی سلول‌های طی‌شده باید دقیقاً برابر عدد T باشه.
۴. مسیر نباید بیش از k سلول منفی داشته باشه.
۵. مسیر نباید بیشتر از L قدم داشته باشه. "یکم دقت کن"
۶. ⚡ مجاز نیستی بیشتر از r بار پشت‌سرهم وارد سلول‌هایی بشی که عددشون e هست. (مثلاً اگه $e = -2$ و $r = 2$ ، نباید سه بار پشت‌سرهم وارد سلول با مقدار -2 بشی.)

🎯 ورودی:

- n
- grid : ماتریس $n \times n$ اعداد صحیح
- T : مجموع انرژی هدف
- k : حداکثر تعداد سلول با انرژی منفی مجاز
- L : حداکثر طول مسیر (یعنی حداکثر تعداد قدم از ابتدا تا انتها)
- e : سلول‌هایی که محدودیت دارند
- r : تعداد محدودیت برای e

✅ خروجی:

- تعداد مسیرهای مجاز که همه‌ی قیود بالا رو رعایت می‌کنن.

با یک DFS همه مسیرهای فضای حالت رو با شروع از (0,0) می‌سازیم و اگه به (n-1,n-1) برسیم و همه محدودیت‌ها برقرار باشن تعدادش رو می‌شماریم. x و y به عنوان موقعیت فعلی، sum به عنوان مجموع انرژی طی‌شده تا اینجا، neg_cnt به عنوان تعداد سلول‌های منفی طی‌شده تا اینجا، steps به عنوان تعداد قدم‌ها تا اینجا و e_cnt به عنوان شمارنده پشت‌سرهم وارد سلول با مقدار e شدن متغیرهای هر حالت هستند که باید ردیابی کنیم.

ورودی L و توضیحش توی صورت سؤال مبهمه. با توجه به حرکت‌های مجاز، طول مسیر معتبر همیشه $2n-2$ است. پس یک چک اولیه می‌کنیم و اگر $L < 2n-2$ بود، هیچ مسیر معتبری وجود نداره و صفر برمی‌گردونیم.

```
if L < 2n - 2:
```

```
    return 0
```

```
valid_cnt = 0
```

```
func BT(x, y, sum, neg_cnt, steps, e_cnt):
```

```
    if x >= n or y >= n:
```

```
        return
```

```
    energy = grid[x][y]
```

```
    sum += energy
```

```
    steps += 1
```

```
    if energy < 0:
```

```
        neg_cnt += 1
```

```
    if energy == e:
```

```
        e_cnt += 1
```

```
    else:
```

```
        e_cnt = 0
```

```
if steps > L or neg_cnt > k or e_cnt > r:
```

```
    return // invalid path
```

```
if x == n-1 and y == n-1 and sum == T:
```

```
    valid_cnt += 1
```

```
    return
```

```
// move right
```

```
BT(x, y + 1, sum, neg_cnt, steps, e_cnt)
```

```
// move down
```

```
BT(x + 1, y, sum, neg_cnt, steps, e_cnt)
```

4023613060 - 4023613068