عددگذاری در آرایه

حل مسئله با برنامه‌ریزی پویا

صورت مسئله

یک آرایه با طول n داریم که هر خانهٔ آن یکی از این دو حالت را دارد:

- یک عدد ثابت بین ۱ تا m
- -۱

باید تعداد حالت‌هایی را بشماریم که بعد از جایگزینی همهٔ -۱ها، برای هر i داشته باشیم:

$$|a_i - a_{i+۱}| \leq ۱$$

هدف، محاسبهٔ تعداد آرایه‌های معتبر با پیچیدگی زمانی $O(nm)$ است.

ایدهٔ اصلی DP

شرط مسئله فقط به دو خانهٔ مجاور وابسته است.
پس اگر مقدار خانهٔ فعلی را بدانیم، فقط کافی است مقدار خانهٔ قبلی را بررسی کنیم.
به همین دلیل از برنامه‌ریزی پویا روی پیشوندهای آرایه استفاده می‌کنیم.
حالت DP را به شکل زیر تعریف می‌کنیم:

$$dp[i][x]$$

یعنی:
تعداد روش‌های معتبر برای پر کردن خانه‌های ۱ تا i به‌طوری که مقدار خانهٔ iام برابر x باشد.

رابطهٔ بازگشتی (Recurrence)

اگر مقدار خانهٔ i برابر x باشد، مقدار خانهٔ قبلی فقط می‌تواند یکی از این سه مقدار باشد:

$$x-۱,\quad x,\quad x+۱$$

زیرا باید شرط زیر برقرار باشد:

$$|a_i - a_{i-۱}| \leq ۱$$

بنابراین رابطهٔ بازگشتی DP برابر است با:

$$dp[i][x] = dp[i-۱][x-۱] + dp[i-۱][x] + dp[i-۱][x+۱]$$

البته فقط مقادیری معتبر هستند که در بازهٔ ۱ تا m قرار داشته باشند.

مقداردهی اولیه و پاسخ نهایی

برای خانهٔ اول دو حالت داریم:

- اگر $a_1 = -1$ باشد:

$$dp[1][x] = 1 \qquad 1 \leq x \leq m$$

- اگر مقدار خانهٔ اول ثابت باشد: فقط همان مقدار برابر ۱ و بقیه برابر صفر خواهند بود.

پس از محاسبهٔ همهٔ حالت‌ها تا خانهٔ n، پاسخ نهایی برابر است با:

$$\sum_{x=1}^{m} dp[n][x]$$

زیرا پاسخ می‌تواند با هر مقدار مجاز در خانهٔ آخر پایان یابد.

شبه کد

```
for x = 1..m:
    if a[1] == -1 or a[1] == x:
        dp[1][x] = 1

for i = 2..n:
    for x = 1..m:

        if a[i] != -1 and a[i] != x:
            continue

        dp[i][x] = dp[i-1][x]

        if x > 1:
            dp[i][x] += dp[i-1][x-1]

        if x < m:
            dp[i][x] += dp[i-1][x+1]

answer = sum(dp[n][x] for x=1..m)
```

اثبات درستی الگوریتم

اثبات را با استقرا روی طول پیشوند انجام می‌دهیم.

فرض استقرا:
فرض کنید $dp[i-۱][x]$ تعداد صحیح روش‌های معتبر برای پر کردن $i-۱$ خانهٔ اول را نشان می‌دهد.

گام استقرا:
برای ساخت حالت‌های مربوط به $dp[i][x]$، تنها حالت‌هایی مجاز هستند که مقدار خانهٔ قبلی یکی از سه مقدار $x-۱$، x یا $x+۱$ باشد.
تمام این حالت‌ها را جمع می‌کنیم، بنابراین:

- هیچ حالت معتبری از قلم نمی‌افتد.
- هیچ حالت نامعتبری شمرده نمی‌شود.
- هر آرایه دقیقاً یک‌بار شمرده می‌شود.

پس رابطهٔ بازگشتی تمام پاسخ‌های صحیح را تولید می‌کند.

پیچیدگی زمانی و حافظه‌ای

برای هر یک از n موقعیت، تمام m مقدار ممکن را بررسی می کنیم.
برای هر حالت نیز فقط سه انتقال بررسی می‌شود.
بنابراین پیچیدگی زمانی برابر است با:

$$O(nm)$$

اگر کل جدول DP را ذخیره کنیم، حافظهٔ موردنیاز:

$$O(nm)$$

خواهد بود.
اما چون هر سطر فقط به سطر قبلی وابسته است، می‌توان تنها دو سطر نگه داشت و حافظه را به:

$$O(m)$$

کاهش داد.