Tutorial: Férias

Arthur Andrade D'Olival

1 Solução do Problema

O problema de planejar as férias de Miguel para minimizar os dias de descanso, respeitando a regra de não repetir a mesma atividade física ou mental em dias consecutivos, pode ser resolvido de forma muito eficiente usando **programação dinâmica**. A intuição é que a decisão do que fazer no dia atual depende exclusivamente da disponibilidade de atividades no dia e do que Miguel escolheu fazer no dia imediatamente anterior.

1.1 Definição do Subproblema

Como a restrição olha apenas para a atividade do dia anterior, precisamos carregar essa informação no nosso estado. Definimos o nosso estado da programação dinâmica como:

$\mathrm{dp[i][j]} = \text{o número mínimo de dias de descanso acumulados até o dia i, dado que a atividade escolhida no dia i foi j.}$

Mapeamos os possíveis valores de j (as escolhas de Miguel) da seguinte forma:

- $j = 0$: Descansar
- $j = 1$: Participar de uma competição
- $j = 2$: Ir à academia

A resposta final para o problema será o menor valor entre todas as atividades possíveis no último dia de férias: $\min(dp[n][0], dp[n][1], dp[n][2])$.

1.2 Função de Transição

Para calcular as opções do dia i, olhamos para os melhores resultados alcançados no dia $i - 1$ e verificamos o que o calendário ($A[i]$) nos permite fazer hoje.

1. Descansar ($j = 0$): Miguel sempre pode descansar em qualquer dia, independentemente do que ele fez ontem. Como descansar adiciona 1 ao nosso contador de dias de descanso, a transição é:

$$dp[i][0] = \min\left(dp[i-1][0], dp[i-1][1], dp[i-1][2]\right) + 1$$

2. Competir ($j = 1$): Miguel só pode competir se houver competição hoje ($A[i] == 1$ ou $A[i] == 3$). Além disso, ele não pode ter competido ontem. Logo, os estados válidos anteriores são apenas descanso (0) ou academia (2):

$$dp[i][1] = \min\left(dp[i-1][0], dp[i-1][2]\right)$$

Se não houver competição hoje, definimos o estado como inalcançável ($dp[i][1] = \infty$).

3. Ir à academia ($j = 2$): De forma análoga, Miguel só pode treinar se a academia estiver aberta ($A[i] == 2$ ou $A[i] == 3$) e se ele não tiver ido à academia ontem. Os estados válidos anteriores são descanso (0) ou competição (1):

$$dp[i][2] = \min\left(dp[i-1][0], dp[i-1][1]\right)$$

Se a academia estiver fechada hoje, definimos o estado como inalcançável ($dp[i][2] = \infty$).

1.3 Casos Base

Para que as transições do primeiro dia de férias ($i = 1$) funcionem corretamente, inicializamos o dia 0 (o momento anterior ao início das férias).

Como antes das férias começarem não há nenhum dia de descanso acumulado, independentemente da atividade "fictícia" que possamos imaginar, zeramos os estados iniciais:

$$dp[0][0] = 0$$
$$dp[0][1] = 0$$
$$dp[0][2] = 0$$