

# Tutorial: Deletar e Ganhar

Arthur Andrade D'Oliveira

O problema pode ser resolvido utilizando uma abordagem clássica de **programação dinâmica**. A ideia central é que, ao escolher um número  $x$ , ganhamos uma certa quantidade de pontos, mas perdemos a possibilidade de escolher os números  $x - 1$  e  $x + 1$ , já que eles devem ser removidos da sequência.

## Modelagem

Primeiro, agrupamos os números iguais da sequência e calculamos o total de pontos que seria obtido caso escolhêssemos todos os elementos de valor  $i$ :

$$\text{points}[i] = (\text{quantidade de vezes que } i \text{ aparece}) \times i$$

Assim, o problema passa a ser equivalente a escolher quais valores  $i$  maximizarão a soma total de pontos, respeitando a restrição de que, ao escolher  $i$ , não podemos escolher  $i - 1$  nem  $i + 1$ .

## Definição da DP

Definimos:

$$dp[i] = \text{pontuação máxima possível utilizando apenas os números de 1 até } i$$

## Função de Transição

A cada passo, temos duas escolhas:

- **Não escolher** o número  $i$ : nesse caso, o resultado é o mesmo que  $dp[i - 1]$ ;
- **Escolher** o número  $i$ : ganhamos  $\text{points}[i]$ , mas não podemos utilizar  $i - 1$ , então somamos com  $dp[i - 2]$ .

A função de transição é, portanto:

$$dp[i] = \max(dp[i - 1], dp[i - 2] + \text{points}[i])$$

## Casos Base

$$dp[0] = 0, \quad dp[1] = \text{points}[1]$$

## Resposta Final

A pontuação máxima possível é:

$$dp[m]$$

onde  $m$  é o maior valor presente na sequência original.

## Complexidade

- **Tempo:**  $O(m)$ , onde  $m$  é o valor máximo na sequência;
- **Espaço:**  $O(m)$ , podendo ser otimizado para  $O(1)$  se armazenarmos apenas os dois últimos estados da DP.