

Aluno: André Santos Rocha

Ra: 235887

Problema 1: C - Fence

O problema fornece um array, no qual o i -ésimo elemento é a altura da i -ésima estaca da cerca. Além disso, temos a entrada de um inteiro k . Devemos imprimir um j tal que a soma das estacas $j, \dots, j+k-1$ seja a menor possível.

1.1 Ideia de solução

Para este problema, a solução será baseada na soma de prefixos. Para cada índice i do array de entrada, guardamos em um vetor `somas` o valor da soma dos elementos que vão de 0 até o índice i em `somas[i]`. Com o vetor `somas` em mãos, computamos agora todas as soluções possíveis. Perceba que, para obter uma solução começando no índice j , fazemos `somas[j+k-1] - somas[j-1]` e armazenamos o resultado em `soluções[j]`. Suponha $V_0 + V_1 + \dots + V_{j-1}$ a soma no índice $j-1$, então no índice $j+k-1$ teremos a soma $V_0 + V_1 + \dots + V_j + \dots + V_{j+k-1}$, subtraindo esses dois valores, nos restará apenas $V_j + V_{j+1} + \dots + V_{j+k-1}$, o que nos dá exatamente k elementos. Por fim, basta percorrer o vetor de soluções e encontrar a menor solução possível, armazenando o índice j .

1.2 Detalhes de implementação

O array de entrada é indexado a partir de 1, ao invés de 0, então é necessário um cuidado maior ao lidar com a entrada e no momento de devolver a resposta.

Problema 1: J - Vlad and a Sum of Sum of Digits

O problema fornece um inteiro n , devemos, para uma sequência de números de 1 a n , substituí-los pela soma de seus dígitos. Depois, precisamos somar a sequência obtida e devolver o valor dessa soma.

1.1 Ideia de solução

Para este problema, a solução será baseada em programação dinâmica. Criamos um vetor `dp` que armazenará nossas soluções. Perceba que `dp[1]` é o próprio 1. Para 2, `dp[2] = soma_digitos(2) + dp[1] = 2 + dp[1] = 3`. E, para i , teremos `dp[i] = dp[i-1] + soma_digitos(i)`, onde `soma_digitos(i)` é uma função que retorna a soma dos dígitos de i . No começo da execução do código, computamos todos os valores até `dp[N]` sendo N o maior valor possível de entrada, o que é no pior caso $O(n)$. E daí, para cada teste, obteremos os valores de resposta em tempo constante.

1.2 Detalhes de implementação

É importante setar $dp[0] = 0$ e $dp[1] = 1$, antes de começarmos a percorrer o vetor de dp a partir de $i = 0$. Só assim, preencheremos o restante dos valores de maneira correta.