Trabalho Prático 3 - Centro de Distribuição

Lucas Almeida Santos de Souza - 2021092563[1]

[1]Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte - MG - Brasil

luscaxalmeidass@ufmg.br

1 Introdução

Este trabalho consiste na resolução de um problema utilizando programação dinâmica. Apresentaremos o problema, sua modelagem, a solução proposta e a análise de complexidade. Por fim, será apresentada uma argumentação para provar que se trata de um problema NP-Completo.

Descrição do problema

O problema em questão envolve um centro de distribuição de ligas metálicas que possui conexões com várias fábricas e clientes, e tem o objetivo de otimizar sua logística. Cada fábrica produz ligas metálicas de tamanhos diferentes, e cada cliente possui uma demanda específica de ligas, medida em metros. O objetivo é minimizar o número de ligas necessárias para atender à demanda, levando em consideração os tamanhos disponíveis nas fábricas.

Por exemplo, se o centro de distribuição tem ligas de tamanhos [1, 5, 10, 20, 25] e um cliente precisa de 40 metros de ligas, o número mínimo de ligas necessárias para suprir essa demanda é 2 (duas ligas de 20 metros cada).

2 Modelagem

A entrada do programa consiste em um arquivo de texto com o seguinte formato:

- A primeira linha contém um inteiro T que representa o número de testes que será feito;
- A segunda linha contém dois inteiros separados por espaço:
  - O primeiro, $1 \leq N \leq 1000$, representa a quantidade de tamanhos de ligas metálicas disponíveis (ou o número de fábricas);
  - O segundo, $1 \leq W \leq 1000000$, representa a demanda do cliente em metros.
- A terceira linha contém uma sequência de inteiros $1 \leq l_i \leq 1000$ separados por espaço, que representam os tamanhos de ligas disponíveis, em metros.

A saída do programa consiste em um arquivo de texto com T linhas, cada uma contendo um inteiro que representa o número mínimo de ligas necessárias para atender à demanda de cada teste.

Implementação

O pseudocódigo da função que resolve o problema pode ser visto abaixo:

minimo(vector l, int w):

```
resultados := [∞] // Vetor de tamanho w + 1 inicializado em ∞
                  // contém o resultado para cada tamanho de encomenda

resultados[0] := 0 // Não é necessário nenhuma liga para uma encomenda de tamanho 0

para cada tamanho de liga l_i faça
    para cada tamanho de encomenda j=1 até w faça
        se l_i ≤ j faça
            resultados[j] = min(resultados[j], resultados[j - l_i] + 1)
        fim se
    fim para cada
fim para cada

retorne resultados[w]
```

O algoritmo apresentado acima utiliza conceitos de programação dinâmica para resolver o problema de modo mais eficiente. O vetor `resultados`, inicializado em ∞, contém o número mínimo de ligas necessárias para atender à demanda de cada tamanho de encomenda. Para cada tamanho de liga disponível menor ou igual à demanda, o algoritmo calcula o número mínimo de ligas necessárias para atender à demanda de tamanho j utilizando a liga de tamanho l_i. O resultado é o mínimo entre o número de ligas necessárias para atender à demanda de tamanho j utilizando a liga de tamanho l_i e o número mínimo de ligas necessárias para atender à demanda de tamanho $j - l_i$ (já calculado anteriormente) mais uma liga de tamanho l_i.

3 Análise de complexidade e NP-Completude

Análise de complexidade

Poderia se pensar inicialmente que o algoritmo apresentado possui complexidade $O(NW)$, onde N é o número de tamanhos de ligas disponíveis e W é a demanda do cliente. Isso porque o algoritmo itera sobre cada tamanho de liga disponível e para cada tamanho de liga disponível itera sobre cada tamanho de encomenda de 1 até W.

No entanto, enquanto o vetor `resultados` afeta a complexidade em seu tamanho N (independente do tamanho dos números que ele contém), a demanda W do cliente afeta a complexidade em sua grandeza, mesmo sendo um só número. Então, uma maneira mais correta de analisar a complexidade do algoritmo é em função do número de bits necessários para representar W em binário. Considerando Y como o número de bits necessários para representar W em binário, então a complexidade do algoritmo é $O(N \cdot 2^Y)$.

NP-Completude

Para provar que o problema é NP-Completo, é necessário reduzir a um problema NP-Completo conhecido. O problema da soma de subconjuntos (Subset Sum) é um problema NP-Completo que consiste em determinar se existe um subconjunto de um dado conjunto cuja soma dos elementos é igual a um determinado valor. Esse problema é bem parecido com o problema das ligas metálicas, mas o problema da soma de soma de conjuntos contabiliza cada valor apenas uma vez, enquanto o problema das ligas metálicas contabiliza cada valor quantas vezes forem necessárias para atender à demanda.

Então, para reduzir o problema das ligas metálicas ao problema da soma de subconjuntos, dado uma entrada $([a_1, a_2, ..., a_n], S)$ basta seguir os seguintes passos para gerar uma entrada $([l_1, ..., l_n, l'_1, ..., l'_n], W)$ para o problema das ligas metálicas:

Seja $b = max(n + 1, S + 1)$, $l_1 = 1$ e para cada $i \in \{1, ..., n\}$, faça:

$$\begin{aligned} l_i &= a_i \cdot b^{n+1} + b^i \\ l'_i &= b^i \end{aligned} \tag{1}$$

E, por fim,

$$W = S \cdot b^{n+1} + \sum_{i=1}^{n} b^i \tag{2}$$

Ao converter os números do subconjunto orginal para a base b desse modo, o problema das ligas metálicas se torna obrigado a escolher uma liga de cada tipo, pois se escolher duas ligas do mesmo tipo, o custo somado será maior do que escolher uma liga de cada tipo. Além disso, o problema das ligas metálicas se torna obrigado a escolher a liga de menor tamanho de cada tipo, pois se escolher uma liga de tamanho maior, o custo somado será maior do que escolher a liga de menor tamanho.

Após essa função, basta analisar o retorno do problema das ligas metálicas. Se o retorno for n, então o problema das ligas metálicas encontrou uma solução que atende à demanda de tamanho W utilizando n ligas, logo existe o subconjunto com soma k do problema da soma de subconjuntos. Se o retorno for ∞, então o problema das ligas metálicas não encontrou uma solução que atende à demanda de tamanho W, logo não existe o subconjunto com soma k. Portanto, o problema das ligas metálicas é NP-Completo.