Trabalho Prático 3
Centro de distribuição

Vinicius Trindade Dias Abel
Matrícula: 2020007112

Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte – MG – Brasil

viniciustda@ufmg.br

1. Introdução

O problema proposto foi desenvolver um algoritmo que determine o menor número de ligas metálicas usadas para suprir uma demanda, levando em conta os tamanhos de liga disponíveis no momento. Para resolver o problema, foi utilizado programação dinâmica.

2. Método

2.1. Estrutura de Dados

A implementação do programa teve como base a estrutura de dados de dois vetores, sendo eles:

- O vetor “tamanhos” que armazena os tamanhos de liga disponíveis; e
- O vetor “resposta” que armazena a quantidade mínima de ligas necessárias para atender a cada demanda de 1 até a demanda de interesse.

2.2. Funções

MinimoLigas

A função recebe como parâmetro um vector<int> tamanhos, que armazena os tamanhos de liga disponíveis, e um int demanda, que é a demanda de interesse do problema.

É declarado um vector<int> resposta, inicializado com 0 em todas posições, para armazenar a quantidade mínima de ligas necessárias para atender a cada demanda de 1 até a demanda de interesse.

Em seguida, o vetor resposta é preenchido realizando a seguinte verificação: se o tamanho da liga disponível for menor ou igual a demanda, calcula o valor mínimo para atender a demanda atual. Este cálculo é realizado com programação dinâmica, pelo bottom up, uma vez que o vetor resposta começa a preencher as demandas pelo caso base que é a demanda 1 e “sobe” até a demanda de interesse.

Após calcular o mínimo de ligas para todo o vetor resposta, sua última posição (resposta[demanda]) é retornada.

Main

A função main recebe o número de testes que serão realizados, e para cada teste, recebe o numero de tamanhos disponíveis, a demanda e, em seguida, os tamanhos disponíveis (que são armazenados no vetor tamanhos). Após receber os tamanhos disponíveis, chama a função MinimoLigas passando o vetor tamanhos e a demanda, imprimindo o resultado.

2.3. Configuração para teste

– Sistema Operacional do computador: Linux Ubuntu;
– Linguagem de programação implementada: C++;
– Compilador utilizado: G++ da GNU Compiler Collection;
– Dados do seu processador: i7;
– Quantidade de memória RAM: 16GB.

3. Análise de Complexidade e NP-Completude

3.1. Análise de Complexidade – tempo de execução no valor numérico da entrada

MinimoLigas

A função tem complexidade O(nm), uma vez que o laço externo tem tamanho n (é a demanda) e o laço interno tem tamanho m (número de tamanhos de liga disponíveis). Entretanto, na maioria dos casos, n é muito maior que m.

Geral

O TP tem complexidade O(lnm). A função main tem um laço externo de tamanho l que também possui um laço interno de tamanho m, além de chamar a função MinimoLigas. Como MinimoLigas predomina sobre o laço de tamanho m, podemos dizer que a complexidade é O(lnm).

3.2. NP-Completude

Problema das ligas
Entrada: Conjunto L, inteiro D
Pergunta: Existe um conjunto de ligas que a soma seja D?

Problema da mochila $\leq_P$ Problema das ligas
(L, D) (L', D')

L – Lista de itens da mochila
D – Capacidade da mochila
L' – Lista dos tamanhos de liga disponíveis
D' – Demanda
Para cada item de L, seu valor e seu peso são iguais ao seu valor.

L’ = L
D’ = D
Executando o algoritmo do problema das ligas, a resposta retornada corresponde ao valor máximo obtido no Problema da mochila.

4. Conclusões

Este trabalho lidou com cálculo do mínimo de ligas necessárias para atender a uma demanda, levando em consideração os tamanhos de liga disponíveis.

Por meio da resolução desse trabalho, foi possível praticar os conceitos relacionados a programação dinâmica.

Durante a implementação da solução do trabalho, houveram importantes desafios a serem superados. Por exemplo, na minha primeira tentativa de resolução, usei a função MinimoLigas de forma recursiva e tive problema na alocação de memória, tentei usar ponteiros para evitar o problema, mas não obtive sucesso. Então decidi mudar a abordagem e não fazer chamadas recursivas, desta forma consegui implementar o algoritmo sem dificuldade.

5. Bibliografia

- Standard C++ Library reference. Disponível em: https://cplusplus.com/reference/

- Comparativo entre a estratégia gulosa e a programação dinâmica para o problema do troco com sistemas de moedas canônicos - Autor: Lucas Vasconcelos Mattioli. Disponível em: https://bdm.unb.br/bitstream/10483/21568/1/2018_LucasVasconcelosMattioli_tcc.pdf

- Relatório de Projeto Supervisionado MS777 Problema do Troco: uma resolução via programação dinâmica - Autor: Gustavo Leite Machado. Disponível em: https://www.ime.unicamp.br/~mac/db/2018-1S-169356.pdf

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., Introduction to Algorithms, 3th ed., MIT Press, Cambridge, Massachusetts, USA, (2009)

6. Instruções para compilação e execução

• Acesse o diretório tp3

• Utilizando o terminal, compile e teste o TP 03 utilizando o comando: make eval

• Com esse comando, será gerado o arquivo tp03.exe e os arquivos *.o