

MC558 - Complexidade de Algoritmos II

Primeiro semestre de 2023

Laboratório 6

O Problema do Troco

Você já parou para pensar o porquê de haver poucas moedas em circulação no comércio? O motivo é que os bancos pagam para as transportadoras de valores transportarem o dinheiro até as agências, e as transportadoras cobram por quilo. Assim, o custo de mil reais em moedas para o banco é muito maior do que o custo de mil reais em notas de cem. Dessa forma, os bancos adquirem poucas moedas e há escassez no mercado.

Pensando em minimizar seus custos com transportes e diminuir a necessidade de moedas pesadas, o banco Unicom passou a exigir que os seus caixas deem sempre o troco mais leve possível. Como comprar equipamentos especializados no Paraguai para todos os caixas sairia muito caro, o banco Unicom contratou você, que cobrou baratinho porque ainda está fazendo graduação, para desenvolver um aplicativo que, dado o valor do troco, devolve a quantidade de cada moeda que deve ser utilizada para formar o troco mais leve.

Formalmente, podemos descrever esse problema da seguinte maneira. Dados n tipos de moedas, tal que $v_i \in \mathbb{Z}^+$ é o valor do i -ésimo tipo de moeda, $p_i \in \mathbb{Z}^+$ o seu peso e $q_i \in \mathbb{Z}^+$ a sua quantidade no caixa, e seja Q um número inteiro positivo que representa o valor do troco, o objetivo é resolver o seguinte modelo:

$$\begin{array}{ll}\text{Minimize} & \sum_i p_i x_i \\ \text{Sujeito a} & \sum_i v_i x_i = Q \\ & x_i \in \{0, 1, \dots, q_i\} \quad \forall i = 1, \dots, n\end{array}$$

Neste laboratório, você deve reduzir o problema do troco para o problema do caminho mínimo em grafos orientados sem arestas de peso negativo. Um exemplo de redução entre problemas que pode lhe auxiliar na solução pode ser encontrado [aqui](#).

Para resolver o problema do caminho mínimo, você irá utilizar como caixa-preta o código disponibilizado (em arquivos auxiliares no SuSy), que contém uma implementação de um algoritmo de caminho mínimo em DAGs (grafos acíclicos dirigidos). O uso dessa implementação é obrigatório e seus arquivos não podem ser alterados. Você deve modificar o arquivo-exemplo t6.c ou t6.cpp para resolver o problema, e submeter apenas esse arquivo no SuSy. O SuSy irá linkar seu código com a implementação fornecida automaticamente.

Pausa para meditação:

No seu programa você deverá implementar uma redução do Problema do Troco para o Problema de Caminho Mínimo em DAGs. Responda (a si mesmo apenas, não é necessário submeter nada!):

O problema dos caminhos mínimos está em P . O problema do troco é NP -difícil e você o reduziu para um problema que está em P . Isto permite concluir que o problema do troco também pertence a P ?

1 Entrada e Saída

Entrada: Na primeira linha da entrada estão dois inteiros n e Q , que indicam, respectivamente, o número de tipos de moedas e o valor do troco que se deseja formar. Você pode supor que $1 \leq n \leq 150$ e que $1 \leq Q \leq 10000$. A seguir vêm n linhas, cada uma contendo três inteiros v , p e q , que indicam, respectivamente, o valor, o peso e a quantidade disponível de um tipo de moeda. Você pode supor que $1 \leq v \leq \min\{150, Q\}$, $1 \leq p \leq 1000$ e $1 \leq q \leq 3$.

Saída: Caso possua solução, a primeira linha deve conter um único número inteiro com o peso total de moedas mínimo necessário. Já se a instância for impossível, imprima o maior valor (menor do que Q) que pode ser obtido com o conjunto de moedas, junto com o peso mínimo para atingir tal valor.

2 Exemplos

Entrada	Saída
4 5	800
4 500 3	
2 400 1	
1 300 1	
3 450 2	

Entrada (impossível)	Saída
4 9	8 50
4 25 2	
6 40 2	
7 30 1	
8 60 3	

3 Implementação e Submissão

- A solução deverá ser implementada em C ou C++11.
- Antes de compilar seu código baixe os arquivos `shortest_path.h` (`shortest_path.hpp`) e `shortest_path.c` (`shortest_path.cpp`) se for usar C (C++) e coloque-os na mesma pasta que sua solução.
- Adicione `shortest_path.c` (`shortest_path.cpp`) ao final do comando de compilação para compilar sua solução em C (C++).
- O programa deve ser submetido no SuSy, com o nome principal **t6** (por exemplo, `t6.c`) e o número máximo de submissões é 20.
- A tarefa contém 10 testes abertos e 10 testes fechados. A nota será proporcional ao número de acertos nos testes fechados.
- **Casos de plágio implicam em nota ZERO na disciplina para todos os envolvidos.**
- **Não é permitido o uso de bibliotecas que não sejam padrão, bem como diretivas ou flags de otimização.**

4 Prazo final de submissão

Segunda-feira 26 de junho às 13h da tarde.