

# CI1238 - Otimização

## Segundo Trabalho Prático

23 de abril de 2024

### 1 Introdução

O trabalho consiste em modelar e implementar, por *Branch & Bound*, o problema *Comissão Representativa*, descrito na Seção 2.

A resolução do problema, ou seja, a descrição do problema, da modelagem e da implementação, deve estar em um texto claro em formato de um artigo em pdf. Além disso, deve ser feita uma análise com duas funções limitantes (*bounds*) diferentes, sendo uma dada pelo professor (Seção 3) e outra escolhida pelos alunos (que deve ser melhor que a dada). A função limitante dos alunos deve ser a default na implementação e a dada pelos professores é escolhida usando uma opção da linha de comando (-a).

O texto deve conter o nome dos autores (alunos), uma introdução com o problema, a modelagem e sua explicação (de por que essa modelagem resolve o problema), detalhes da implementação (com exemplos de uso), e uma análise do uso das funções limitantes. Nesta análise devem ser feitas e comparadas contagens de número de nós da árvore e tempo de execução (com relatório gerado pelo programa). Outras métricas também podem ser usadas.

Todas as referências que forem usadas devem estar citadas corretamente no texto.

Para facilitar a análise, o seu programa deve ter a opção de fazer ou não os cortes por viabilidade e otimalidade. Ou seja, com a opção de linha de comando -f os cortes por viabilidade não devem estar ativos; e com a opção de linha do comando -o os cortes por otimalidade são desligados.

Você pode usar bibliotecas para estruturas de dados (como listas, conjuntos etc), mas não para o algoritmo de resolução principal do problema. O seu programa deve compilar e executar nas servidoras do DINF.

O trabalho deve ser entregue com um **makefile** de forma que ao digitar o comando **make** o executável **comissao** seja construído.

Resumindo, o texto deve ter:

- identificação;
- explicação do problema;
- modelagem;
- análise das funções limitantes;
- detalhes da implementação.

A implementação:

- deve ter executável de nome **comissao**;
- na opção default todos os cortes (viabilidade e otimalidade) estão ativos e a função limitante é a criada pelos autores;
- com a opção **-f** na linha de comando deve desligar os cortes de viabilidade;
- com a opção **-o** na linha de comando deve desligar os cortes de otimalidade;
- com a opção **-a** na linha de comando deve usar a função limitante dada pelos professores;
- deve gerar relatório (na saída de erro padrão, **stderr**) com número de nós da árvore e tempo gasto (sem contar o tempo de entrada e saída).

Você deve entregar um arquivo compactado (**tar.gz**) com os seguintes arquivos no diretório corrente:

- texto (em pdf);
- os fontes (podem estar em subdiretórios);
- makefile;
- exemplos usados na análise (podem estar em subdiretórios).

A entrega deve ser feita por e-mail para **andre@inf.ufpr.br** em um arquivo compactado com todos os arquivos do trabalho, com assunto “Otimização-trabalho 2” (exatamente).

## 2 O problema

### Comissão Representativa

Um órgão do governo quer montar uma comissão para tratar de um certo assunto. Mas este órgão, preocupado com a representatividade da comissão, quer que todos os grupos (previamente elencados) da sociedade sejam representados. Uma pessoa pode fazer parte de mais de um destes grupos, portanto podemos ter menos representantes que grupos. Dados um conjunto  $S$  de grupos, um conjunto  $C$  de candidatos, e um subconjunto  $S_c \subseteq S$ , para cada candidato  $c$ , indicando os grupos dos quais  $c$  faz parte, devemos encontrar um conjunto  $X \subseteq C$  tal que  $\cup_{c \in X} S_c = S$  e  $|X|$  seja mínimo.

### 2.1 Formato de entrada e saída

Os formatos de entrada e saída, são descritos a seguir e devem ser usados a entrada e a saída padrão (`stdin` e `stdout`).

A entrada é formada de um conjunto de números inteiros. Os números podem estar separados por 1 ou mais espaços, tabs ou fim de linha.

**Entrada:** Inicia com os valores de  $\ell = |S|$  e  $n = |C|$  na primeira linha (separados por espaço). Considere que  $S = \{1, \dots, \ell\}$  e  $C = \{1, \dots, n\}$ . Em seguida temos  $n$  blocos (um para cada candidato). Um bloco de candidato começa com um valor,  $s$ , indicando o número de grupos da sociedade que ele faz parte. Em seguida temos  $s$  números, que são os índices dos grupos.

**Saída:** Os números do conjunto  $X$ , em uma mesma linha, separados por espaço (simples) e sem espaço no começo nem no fim da linha, ordenados em ordem crescente.

Caso a instância não tenha solução viável, informe isso escrevendo “Inviavel” (sem acento) na saída.

### 2.2 Exemplos

Exemplos de entrada e saída.

#### 2.2.1 Exemplo simples com $\ell = 2$ e $n = 3$

**Entrada:**

2 3  
1 1  
1 2  
2 1 2

**Saída:**

3

### 2.2.2 Exemplo inviável

**Entrada:**

2 3  
1 1  
1 1  
1 1

**Saída:**

Inviavel

## 3 Função limitante dada

Dados os candidatos já escolhidos ( $E$ ) e o conjunto de candidatos ainda por escolher ( $F$ ), podemos definir a função  $B_{dada}(E, F)$  por:

$$B_{dada}(E, F) = |E| + \begin{cases} 0, & \text{se } \ell = |\cup_{c \in E} S_c|, \\ 1, & \text{se } \ell > |\cup_{c \in E} S_c|. \end{cases}$$

Ou seja,  $B_{dada}(E, F)$  é o tamanho de  $E$  mais 1 (um) caso existam grupos ainda não atendidos.

## 4 Dicas

Para fazer o *Branch & Bound* é preciso escolher uma estratégia de ramificação. Esta estratégia deveria (embora não seja essencial) só passar por soluções (parciais) viáveis.

Lembrem que uma função limitante em um problema de minimização deve ser sempre menor ou igual ao valor ótimo do subproblema, para garantir que nenhuma solução ótima seja cortada.