

## Atividade 1 – Preprocessamento para instâncias do problema de Set Covering.

---

# 1 Descrição

Esta atividade consiste em escrever um programa em C++ que gera um arquivo contendo uma modelo de programação linear inteira para uma dada instância do problema de *Set Covering*.

O modelo resultante deve seguir rigosamente o formato aqui descrito, caso contrário nenhuma pontuação será concedida. Além disso, o programa deve realizar certas simplificações no problema antes de gerar o modelo, conforme visto em sala de aula e descrito brevemente neste documento.

## 1.1 Formato dos dados de entrada

Seu programa deve ler um arquivo chamado `entrada.txt`. Se o material submetido pela equipe como resposta exigir a mudança do código-fonte para que o nome do arquivo seja especificado, ou exigir a entrada do nome do arquivo via teclado, o material não será corrigido e receberá crédito zero.

O arquivo de entrada será do tipo texto. A primeira linha do arquivo conterá um inteiro, informando o número de objetos (chamarei esse inteiro de  $m$  no restante deste documento). A segunda linha conterá um inteiro, informando o número de subconjuntos de  $\{1, \dots, m\}$  (chamarei esse inteiro de  $n$  no restante deste documento). Cada uma das  $n$  linhas seguintes conterá um dos  $n$  subconjuntos. Cada linha será composta de um ou mais números inteiros, separados uns dos outros por um espaço em branco. Os números inteiros presentes nessas linhas serão valores pertencentes ao conjunto  $\{1, \dots, m\}$ . Abaixo, forneço um exemplo de arquivo de entrada.

```
5
5
1 2
2 3 5
1 3 4
1 2 5
3
```

O problema de *Set Covering* a ser modelado é aquele visto em sala de aula, e repetido a seguir para sua conveniência. Os  $n$  subconjuntos serão denotados a partir de agora por  $S_1, S_2, \dots, S_n$ .

$$\begin{array}{ll} \min & \\ & \sum_{j=1}^n x_j \end{array} \quad (1)$$

$$\begin{array}{ll} \text{s.a} & \\ & \sum_{\substack{j=1 \\ i \in S_j}}^n x_j \geq 1, \quad i = 1, \dots, m \end{array} \quad (2)$$

$$x_j \in \{0, 1\}, \quad j = 1, \dots, n. \quad (3)$$

## 1.2 Formato dos dados de saída

A saída de seu programa deve ser um arquivo do tipo texto, chamado `saida.txt`. Variações do nome ou a necessidade de digitar o nome levarão a crédito zero na atividade.

O arquivo deve conter na primeira linha a palavra “Minimize”. A segunda linha deve conter a cadeia “obj: ”, seguida da soma das variáveis, indicando a função objetivo do problema. O nome de cada variável seguirá o formato “ $x_j$ ”, com  $j$  indicando o índice da variável, que variará entre 1 e  $n$ . A próxima linha deve conter a cadeia “Subject To”.

A partir deste ponto, o arquivo deve conter uma linha para restrição do modelo. Cada linha deve iniciar com um identificador do objeto correspondente, no formato “ $c_i$ : ”, com  $i$  indicando o índice do objeto. A seguir, a linha deve conter uma soma das variáveis envolvidas na restrição, sendo esta soma seguida da cadeia “ $\geq 1$ ”.

A linha seguinte deve conter a cadeia “Bounds”. Cada uma das  $n$  linhas seguintes deve informar os limites inferior e superior de uma das variáveis, preferencialmente na ordem de  $x_1$  a  $x_n$ . O formato será um valor numérico (0 ou 1), seguido de “ $\leq$ ”, do nome da variável, do símbolo “ $\leq$ ” novamente, e de outro valor numérico (0 ou 1).

A linha seguinte deve conter a cadeia “General”. A linha seguinte deve conter simplesmente uma lista de todas as variáveis do problema, separadas por um espaço em branco. A última linha deve conter a cadeia “End”.

A seguir, mostro um exemplo de arquivo neste formato, construído a partir dos dados de entrada da subseção anterior. Atente para o fato de que **este arquivo não é uma saída correta para esta atividade**, porque não leva em conta as simplificações que são descritas na próxima subseção. Ele é fornecido aqui apenas como um exemplo do formato de saída.

```
Minimize
  obj: x1 + x2 + x3 + x4 + x5
Subject To
  c1: x1 + x3 + x4 >= 1
  c2: x1 + x2 + x4 >= 1
  c3: x2 + x3 + x5 >= 1
  c4: x3 >= 1
  c5: x2 + x4 >= 1
Bounds
  0 <= x1 <= 1
  0 <= x2 <= 1
  0 <= x3 <= 1
  0 <= x4 <= 1
  0 <= x5 <= 1
General
  x1 x2 x3 x4 x5
End
```

### 1.3 Simplificações

Seu programa deve necessariamente realizar todas as simplificações descritas nesta seção.

Se um objeto pertencer a apenas um dos subconjuntos  $S_j$ , então a variável  $x_j$  deve necessariamente assumir valor 1 em qualquer solução do problema. Seu programa deve detectar todas estas situações e especificar na seção “Bounds” que o limite inferior de cada uma destas variáveis é 1. Em

nosso exemplo, a variável  $x_3$  enquadra-se nessa situação.

Perceba que o fato de que  $x_3 = 1$  implica na satisfação das restrições correspondentes aos objetos 1, 3, e 4, uma vez que  $x_3$  participaria destas restrições. Seu programa deve detectar todas as restrições com esta característica e não deve escrevê-las no arquivo de saída, afinal a satisfação de tais restrições já está assegurada.

Por fim, levando em conta as restrições que sobram após tais simplificações, seu programa deve detectar todos os casos em que uma restrição é *implícada* por outra. Em nosso caso, a restrição correspondente ao objeto  $i$  é implicada pela restrição correspondente ao objeto  $k$  se as variáveis presentes na restrição correspondente ao objeto  $k$  estão todas presentes na restrição correspondente ao objeto  $i$ . Em nosso exemplo, temos o caso em que a restrição  $c2$  é implicada pela restrição  $c4$ .

O arquivo a seguir mostra o que seria o resultado final **correto** da execução do programa sobre o arquivo de entrada utilizado aqui como exemplo.

```
Minimize
  obj: x1 + x2 + x3 + x4 + x5
Subject To
  c5: x2 + x4 >= 1
Bounds
  0 <= x1 <= 1
  0 <= x2 <= 1
  1 <= x3 <= 1
  0 <= x4 <= 1
  0 <= x5 <= 1
General
  x1 x2 x3 x4 x5
End
```

## 2 Entrega

- Pontuação da atividade: 1,5 (um ponto e meio) na primeira nota. Essa pontuação não é extra.
- Entregável: código-fonte comentado e funcional em C++ padrão.
- Data de entrega: 10 de março de 2020, 23:59:59, via SIGAA.
- **Não será aceito envio via e-mail ou fora do prazo.** O prazo é dimensionado para incluir 48h a mais do que uma estimativa realista para resolução da atividade. Para evitar surpresas, como interrupções de conectividade, você deve considerar submeter sua resolução com antecedência.

## 3 Mais detalhes

- A atividade pode ser feita individualmente ou em dupla. **A definição dos membros da equipe deve ser feita via SIGAA, com a opção “Cadastrar grupo”.** Não utilize o código-fonte, nem mensagem via SIGAA, para comunicar esta informação. Os comentários no formulário de envio do SIGAA não serão levados em conta.
- Se for necessário submeter múltiplos arquivos, compacte-os em um só arquivo antes do envio.