

# CI1238/CI7056

## Otimização/Tópicos em Algoritmos

Segundo Trabalho

26 de junho de 2021

### 1 Introdução

O trabalho consiste em modelar e implementar, por *Branch & Bound*, o problema *Caminhada Máxima*, 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 sua função limitante (*bound*).

O texto deve conter o nome do autor (aluno), 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 da função limitante. Nesta análise devem ser feitas 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.

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 **caminhada** seja construído.

Resumindo, o texto deve ter:

- identificação;
- explicação do problema;
- modelagem;

- análise da função limitante;
- detalhes da implementação.

A implementação:

- deve ter executável de nome **caminhada**;
- 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**, preferencialmente em um arquivo compactado com todos os arquivos do trabalho, com assunto “Otimização-trabalho 2” (exatamente).

## 2 O problema

### **Caminhada máxima**

Toda manhã Sr. Gump sai para caminhar por sua cidade. Ele não gosta de passar mais de uma vez por cada lugar. Mas ele gosta de demorar em seu passeio. Dado um conjunto de lugares (vértices), as ligações entre estes lugares (arestas) e os custos (tempo) de percorrer cada uma destas ligações, devemos encontrar um trajeto para o Sr. Gump que seja o mais longo possível, sem repetir lugares.

### 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 um ou mais espaços, tabs ou fim de linha.

A saída também é um conjunto de números inteiros. Nas linhas com mais de número, estes números devem estar separados por apenas um espaço e sem espaço no começo nem no fim da linha.

**Entrada:** Inicia com o número de lugares,  $n$ . Em seguida temos  $n - 1$  linhas, representando uma matriz simétrica  $n \times n$ ,  $A = [a_{ij}]$ , com as ligações e os pesos. Essa matriz está representada somente pelos valores  $a_{ij}$  com  $i < j$ . Ou seja, a primeira linha tem  $n - 1$  números, representando os valores de  $a_{12}, a_{13}, \dots, a_{1n}$ ; a segunda linha tem os valores de  $a_{23}, \dots, a_{2n}$ ; a linha  $i$  tem os valores  $a_{i(i+1)}, \dots, a_{in}$ . Os valores da diagonal principal são 0 (ou seja,  $a_{ii} = 0$ , para todo  $1 \leq i \leq n$ ). Se  $a_{ij} = 0$ , com  $i \neq j$  (fora da diagonal principal) então não existe a ligação entre o lugar  $i$  e o lugar  $j$ . O tempo de percorrer uma ligação entre os lugares  $i$  e  $j$  é dado por  $a_{ij}$ . O lugar inicial é o lugar 1.

**Saída:** Uma linha com o valor do tempo total da caminhada máxima. Em seguida uma linha com a sequência de lugares percorridos. O lugar 1 deve ser o primeiro e o último.

## 2.2 Exemplos de entrada e saída

Os relatórios com o número de nós e o tempo foram omitidos nestes exemplos, já que podem variar com a modelagem e a função limitante escolhida.

### 2.2.1 Exemplo simples com $n = 4$

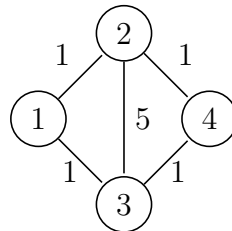


Figura 1: Desenho do exemplo 2.2.1

**Entrada:**

```

3
1 1 0
5 1
1

```

**Saída:**

```

7
1 2 3 1

```

**2.2.2 Exemplo com  $n = 6$  com partes inacessíveis (pois não tem caminho de volta)**

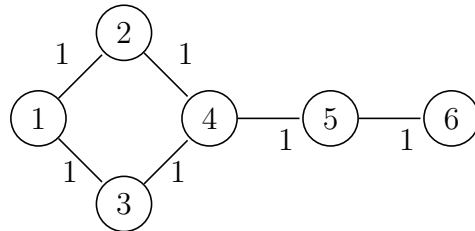


Figura 2: Desenho do exemplo 2.2.2

**Entrada:**

```

6
1 1 0 0 0
0 1 0 0
1 0 0
1 0
1

```

**Saída:**

```

4
1 2 4 3 1

```