



Universidade Federal de Viçosa – Campus UFV-Florestal  
Ciência da Computação – Projeto e Análise de Algoritmos  
Professor: Daniel Mendes Barbosa

## Trabalho Prático 2

Este trabalho é **obrigatoriamente em grupo**. Os grupos já foram definidos [nesta planilha](#) e este trabalho deverá ser entregue no PVANet Moodle de acordo com as instruções presentes no final da especificação.

Na galáxia XX56-000 reside Lancelot Alfredo II, um jovem E.T que possui grandes responsabilidades. Seu pai, Arthur IV é o comandante dessa galáxia e deu ao seu herdeiro uma missão de ir a outra galáxia nomeada XX57-000 entregar uma mensagem criptografada

“TGFuY2Vsb3QsIGRpZ2FtIGegZWxlcyBxdWUgc8OjbyBkb2lzIHByb2JsZW1hcy4gTsOjbyB0ZW0gY29tbyByZXNvbHJlciBvcyAyIGVtIDEu”. No entanto, Alfredo II não sabe quantos caminhos com **custo mínimo** existem para viajar. Por isso, o filho de Arthur IV deseja planejar sua viagem com sua equipe de estrategistas e matemáticos.

Assim, vocês precisam implementar, utilizando **programação dinâmica**, um algoritmo que informe ao Lancelot Alfredo II e gerações futuras quantos caminhos com custo mínimo existem entre viagens intergaláticas. A viagem pode ser representada por um grid de  $N$  linhas por  $M$  colunas, a viagem da galáxia XX56-000 para a galáxia XX57-000 possui 3 linhas e 3 colunas, conforme mostra a figura abaixo.

1	3	1
1	5	1
4	2	1

Existe apenas 1 caminho mínimo possível como ilustrado acima. **O Lancelot Alfredo II sempre sairá do campo (1,1) com o destino em (N, M). Como a nave é antiga, ela apenas se movimenta para baixo e para o lado direito.**

O problema pode ser descrito formalmente da seguinte forma:

Existe uma matriz com  $N$  linhas horizontais e  $M$  colunas verticais. Assuma que a tupla  $(i, j)$  define um campo na  $i$ -ésima linha e na  $j$ -ésima coluna. Para cada  $i$  e  $j$

$(1 \leq i \leq N, 1 \leq j \leq M)$ , o campo  $(i, j)$  é descrito pelo número  $a_{i,j}$ . É garantido que  $a_{i,j}$  é positivo.

**Determine a quantidade de caminhos mínimos possíveis começando do campo  $(1, 1)$  e terminando no campo  $(N, M)$ , com movimentos apenas para direita e para baixo.**

### Restrições

- $N$  e  $M$  são inteiros
- $2 \leq N, M \leq 100$
- $a_{i,j} \geq 0$
- Os campos  $(1, 1)$  e  $(N, M)$  são campos sem asteroides

Seu programa deverá obrigatoriamente usar **somente programação dinâmica**.

Importante:

- você deverá definir as estruturas de dados necessárias ao algoritmo;
- na **documentação** você deverá explicar seu algoritmo com base nos conceitos de programação dinâmica, e como ele foi implementado;

### Entrada

O arquivo terá um formato padronizado, sendo que na primeira linha do arquivo temos, separados por espaços: a quantidade de linhas e a quantidade de colunas.

Após isso, será descrito o caractere de cada campo conforme abaixo:

$N \ M$

$a_{1,1}, a_{1,2}, \dots, a_{1,m}$

.

.

.

$a_{n,1}, a_{n,2}, \dots, a_{n,m}$

O exemplo abaixo representa a viagem intergalática da figura anterior. Como indicado pela primeira linha, a matriz possui 3 linhas e 3 colunas.

## Caso de Teste 1

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

## Saída

Os resultados do algoritmo devem ser, no mínimo, apresentados na saída padrão de acordo com as seguintes especificações. Note que você deve mostrar o custo mínimo e a quantidade de caminhos que existem com esse custo mínimo.

Segue a solução para os exemplos de entrada:

```
Soma Mínima: 7
Quantidade de Caminhos: 1
```

## Interface

O programa não precisa ter interface. É suficiente que ele receba por linha de comando o caminho do arquivo contendo o mapa da viagem intergalática, mostrando em seguida os resultados na saída padrão. Considere que os arquivos de entrada seguirão fielmente o formato que foi definido.

## Tarefas extras

**Tarefas que podem ser feitas além do básico (que podem servir inclusive como diferencial no ranqueamento das notas dos trabalhos):**

1. Plotar gráficos de desempenho de tempo do algoritmo para diferentes tamanhos de entrada, definidos pelo grupo.

- Faça exatamente o que está sendo pedido neste trabalho, ou seja, mesmo que o grupo tenha uma ideia mais interessante para o programa, deverá ser implementado exatamente o que está definido aqui no que diz respeito ao problema em si e ao paradigma programação dinâmica. No entanto, o grupo pode implementar algo além disso, desde que não atrapalhe a obtenção dos resultados necessários a esta especificação.

## Caso de Teste 2

[illegible]

```
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
```

```
Soma Mínima: 19
Quantidade de Caminhos: 48620
```

### Caso de Teste 3

```
2 3
1 2 3
4 5 6
```

```
Soma Mínima: 12
Quantidade de Caminhos: 1
```

### Formato e data de entrega

Os arquivos com o código-fonte (projeto inteiro do Codeblocks ou arquivos .c, .h e makefile), juntamente com um arquivo PDF (**testado, para ver se não está corrompido**) contendo a **documentação**. A documentação deverá conter:

- explicação do algoritmo projetado;
- implementação do algoritmo projetado (estruturas de dados criadas, etc);
- resultados de execução, mostrando entrada e saída;
- arquivos de entrada usados nos testes;
- explicação de como compilar o programa.

Mais direcionamentos sobre o formato da documentação podem ser vistos no documento [“Diretrizes para relatórios de documentação”](#).

**Importante:** Entregar no formato **ZIP**. As datas de entrega estarão configuradas no PVANet Moodle. É necessário que apenas um aluno do grupo faça a entrega, mas o PDF da documentação deve conter os nomes e números de matrícula de todos os alunos que participaram efetivamente em sua capa ou cabeçalho.

Bom trabalho!