

Trabalho Prático 2 - Número de Caminhos

Problema

Joãozinho mora em Johnville, uma cidade projetada tendo como modelo a ilha de Manhattan, onde fica localizada a cidade de New York. Esta cidade tem um traçado de ruas e avenidas em forma de grade (ver figura 1). As ruas são sempre horizontais e as avenidas são verticais. Porém, algumas esquinas são fechadas. A esquina fechada está representada como um quadrado cinza sobre a referida esquina, como pode ser visto na figura 1. Ajude Joãozinho a determinar o número de diferentes caminhos mais curtos a partir da esquina A para uma esquina B em Johnville. O ponto A sempre será a primeira linha e coluna, como mostrado na figura 1.

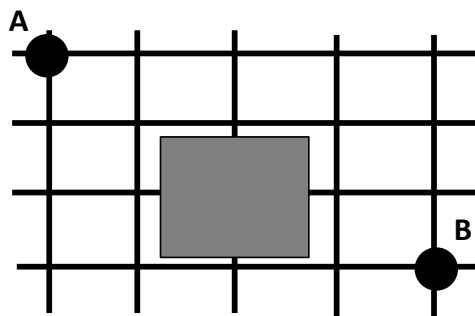


Figura 1: Grade da cidade.

Entrada e Saída

A entrada de dados irá especificar as tarefas a serem realizadas. Cada tarefa é definida por uma cidade e diversos destinos para o caminho mais curto (esquina B).

Mais especificamente, para cada tarefa, o arquivo de entrada contém o tamanho da cidade, as esquinas bloqueadas e as esquinas B. Para cada esquina B, o programa deverá determinar quantos caminhos mais curtos há entre a esquina A e a esquina B.

A entrada será lida de um arquivo e o resultado do programa deve ser impresso em arquivo de saída diferente do arquivo de entrada. Ambos arquivos devem ser passados por parâmetro na chamada do executável:

```
./tp2 input.txt output.txt
```

O arquivo de entrada possui um inteiro N na primeira linha onde N é o número de tarefas a serem computadas. Em seguida, cada uma das N tarefas são definidas da seguinte forma.

A primeira linha possui dois inteiros XY que indicam as 2 dimensões da cidade. A próxima linha indica quantas esquinas E estão bloqueadas, seguida de E s linhas que indicam as posições das esquinas fechadas com o número da avenida (x) e rua (y). Depois, temos o número C de esquinas B, seguido de C s linhas que indicam as posições das esquinas B com o número da avenida (x) e da rua (y).

Para cada esquina B, deve ser impresso no arquivo de saída, a resposta (número de caminhos mais curtos). Entre saídas das tarefas, deve ser incluída uma linha em branco separando as saídas.

Exemplo

A seguir temos um exemplo de funcionamento do programa. A primeira cidade é a mesma da figura 1.

Entrada:

```
2
4 5
1
3 3
2
1 5
3 4
2 2
0
1
2 2
```

Saída:

```
1
4

2
```

Entrega

- A data de entrega desse trabalho é **20 de Outubro**.
- A penalização por atraso obedece à seguinte fórmula $2^{d-1}/0.32\%$, onde d são os dias úteis de atraso.
- Submeta apenas um arquivo chamado `<numero_matricula>_<nome>.zip`. Não utilize espaços no nome do arquivo. Ao invés disso utilize o caractere `'_'`.
- Não inclua arquivos compilados ou gerados por IDEs. **Apenas** os arquivos abaixo devem estar presentes no arquivo zip.
 - Makefile
 - Arquivos fonte (*.c e *.h)
 - Documentacao.pdf
- Não inclua **nenhuma pasta**. Coloque todos os arquivos na raiz do zip.
- Siga rigorosamente o formato do arquivo de saída descrito na especificação. Tome cuidado com whitespaces e formatação dos dados de saída
- **NÃO SERÁ NECESSÁRIO ENTREGAR DOCUMENTAÇÃO IMPRESSA!**
- Será adotada **média harmônica** entre as notas da **documentação e da execução**, o que implica que a nota final será 0 se uma das partes não for apresentada.

Documentação

A documentação não deve exceder 10 páginas e deve conter pelo menos os seguintes itens:

- Uma **introdução** do problema em questão.
- **solução proposta** Explique como representou as estruturas de dados.
- **Análise de complexidade** de tempo e espaço da solução implementada.

Código

- O código deve ser obrigatoriamente escrito na **linguagem C**. Ele deve compilar e executar corretamente nas máquinas Linux dos laboratórios de graduação.
- O utilitário ***make*** deve ser utilizado para auxiliar a compilação, um arquivo *Makefile* deve portanto ser incluído no código submetido.
- As estruturas de dados devem ser **alocadas dinamicamente** e o código deve ser **modularizado** (divisão em múltiplos arquivos fonte e uso de arquivos cabeçalho .h)
- **Variáveis globais** devem ser evitadas.
- Parte da correção poderá ser feita de forma automatizada, portanto **siga rigorosamente os padrões de saída especificados**, caso contrário sua nota pode ser prejudicada.
- **Legibilidade e boas práticas** de programação serão avaliadas.