

INSTITUTO SUPERIOR TÉCNICO

Análise e Síntese de Algoritmos

2022/2023

1º Projecto

Data enunciado: 25 de novembro de 2022
Data Limite de Entrega: 13 de dezembro de 2022 (18h00)

Descrição do Problema

Pretende-se ladrilhar, ou seja cobrir com ladrilhos, uma dada área definida sobre um retângulo, delimitada por um caminho em escada, tal como ilustrado na Figura 1. A área a ladrilhar é definida sobre uma grelha unitária, com n linhas e m colunas. Os ladrilhos a utilizar são sempre quadrados com dimensões do lado múltiplas da unidade, ou seja: 1×1 , 2×2 , etc. Considerando que pode utilizar um número arbitrário de ladrilhos quadrados das várias dimensões, existem diversas configurações possíveis dos ladrilhos que permitem cobrir totalmente a área em questão.

No exemplo ilustrado na Figura 1, existem 4 configurações possíveis para ladrilhar a área em questão. Uma configuração apenas utiliza ladrilhos com dimensão 1×1 , sendo que as restantes três utilizam ladrilhos com dimensões 1×1 e 2×2 .

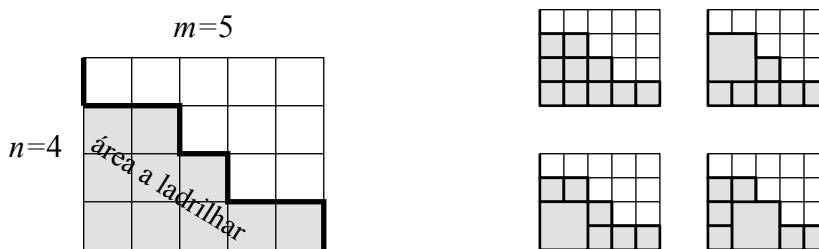


Figura 1: Exemplo de uma área a ladrilhar e respetivas configurações.

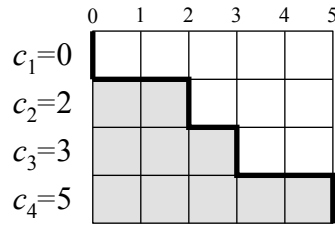


Figura 2: Especificação de um caminho em escada pelos valores c_i , para $n = 4$ e $m = 5$.

Objectivo: Dadas as dimensões n e m , bem como a especificação do caminho em escada que delimita a área a ladrilhar, pretende-se determinar o **número de configurações distintas** de ladrilhos que permitem ladrilhar a referida área.

Tal como ilustrado na Figura 2, a especificação do caminho em escada será efetuada através de uma sequência de n valores inteiros c_i , para $i = 1, 2, \dots, n$, que indicam o índice da coluna pela qual passa o caminho na i -ésima linha. Note que $0 \leq c_i \leq m$, para $i = 1, 2, \dots, n$. Note também que $c_i \geq c_{i-1}$, para $i = 2, 3, \dots, n$. A Figura 3 ilustra todos os caminhos em escada possíveis para $n = 3$ e $m = 3$.

Input

O ficheiro de entrada contendo a informação relativa ao problema a resolver, é definido da seguinte forma:

- uma linha contendo apenas um valor inteiro n , que indica o número de linhas da grelha unitária utilizada para descrever a área a ladrilhar;
- uma linha contendo apenas um valor inteiro m , que indica o número de colunas da grelha unitária utilizada para descrever a área a ladrilhar;
- n linhas, em que cada linha contém apenas um valor inteiro c_i , que indica o índice da coluna pela qual o caminho em escada passa na i -ésima linha.

Output

O programa deverá escrever no output uma linha contendo um único valor inteiro, que corresponderá ao número de configurações distintas de ladrilhos que permitem ladrilhar a área em questão. Considere que, para o caso em que o caminho em escada é definido por $c_i = 0$, para $i = 1, 2, \dots, n$, o número de configurações é 0.

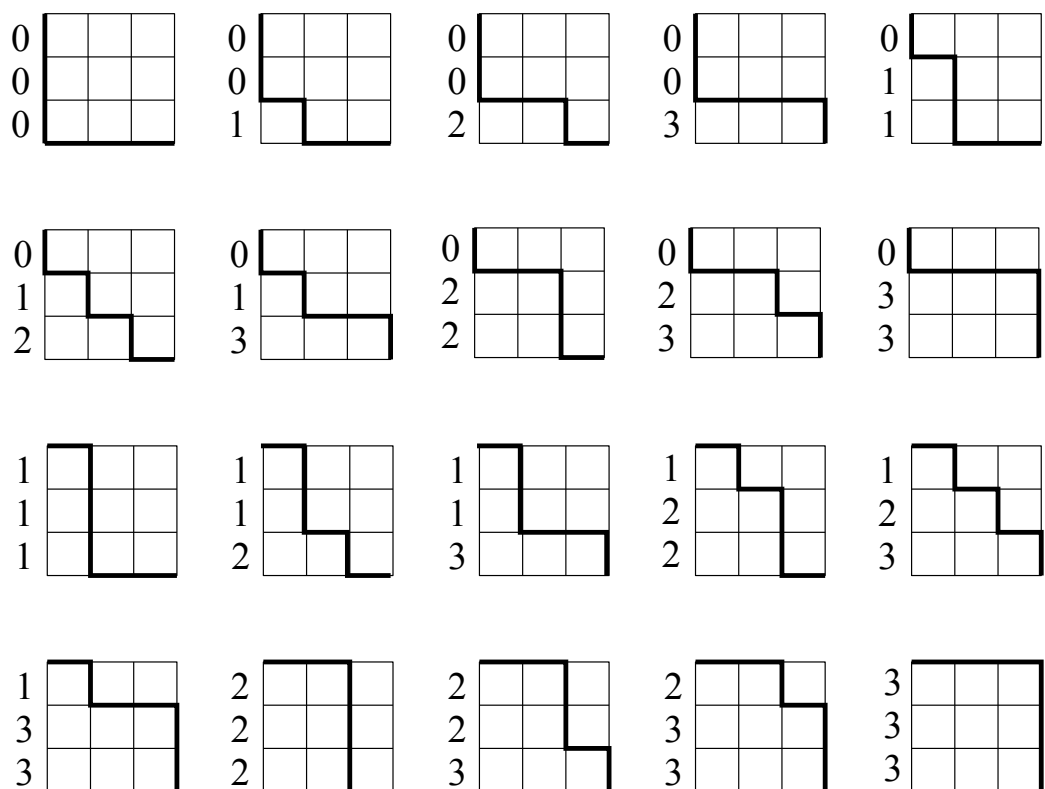


Figura 3: Ilustração de todos os caminhos em escada possíveis para $n = 3$ e $m = 3$, com os respectivos valores de c_i .

Exemplos

Input

4
5
0
2
3
5

Output

4

Input

3
4
4
4
4

Output

13

Implementação

A implementação do projecto deverá ser feita preferencialmente usando as linguagens de programação C++ ou C. Submissões nas linguagens Java/Python também serão aceites, embora fortemente desaconselhadas. Alunos que o escolham fazer devem estar cientes de que submissões em Java/Python podem não passar todos os testes mesmo implementando o algoritmo correcto de forma eficiente.

O tempo necessário para implementar este projecto é inferior a 15 horas.

Parâmetros de compilação:

```
C++: g++ -O3 -std=c++11 -Wall file.cpp -lm
C: gcc -O3 -std=c11 -Wall file.c -lm
Javac: javac File.java
Java: java -classpath . File
Python: python3 file.py
```

Submissão do Projecto

A submissão do projecto deverá incluir um relatório resumido e um ficheiro com o código fonte da solução. Informação sobre as linguagens de programação possíveis está disponível no website do sistema Mooshak. A linguagem de programação é identificada pela extensão do ficheiro. Por exemplo, um projecto escrito em C deverá ter a extensão .c. Após a compilação, **o programa resultante deverá ler do standard input e escrever para o standard output**. Informação sobre as opções e restrições de compilação podem ser obtidas através do botão help do sistema Mooshak. O comando de compilação não deverá produzir output, caso contrário será considerado um erro de compilação.

Relatório: deverá ser submetido através do sistema Fénix, no formato PDF, com não mais de 2 páginas, fonte de 12pt, e 3cm de margem. O relatório deverá incluir uma descrição da solução, a análise teórica e a avaliação experimental dos resultados. O relatório deverá incluir qualquer referência que tenha sido utilizada na realização do projecto. Relatórios que não sejam entregues em formato PDF terão nota 0. Atempadamente será divulgado um template do relatório.

Código fonte: deverá ser submetido através do sistema Mooshak. O código fonte será avaliado automaticamente pelo sistema Mooshak (<http://acp.tecnico.ulisboa.pt/~mooshak/>). Os alunos são encorajados a submeter, tão cedo quanto possível, soluções preliminares para o sistema Mooshak. Note que apenas a última submissão será considerada para efeitos de avaliação. Todas as submissões anteriores serão ignoradas.

Avaliação

O projecto deverá ser realizado em grupos de um ou dois alunos e será avaliado em duas fases. Na primeira fase, durante a submissão, cada implementação será executada num conjunto de testes, os quais representam 85% da nota final. Na segunda fase, o relatório será avaliado. A nota do relatório contribui com 15% da nota final.

Avaliação Automática

A primeira fase do projecto é avaliada automaticamente com um conjunto de testes, os quais são executados num computador com o sistema operativo **GNU/Linux**. É essencial que o código fonte compile sem erros e respeite os standards de entrada e saída indicados anteriormente. Os projectos que não respeitem os formatos especificados serão penalizados e poderão ter nota 0, caso falhem todos os testes. Os testes **não serão divulgados antes da submissão**. No entanto, todos os testes serão disponibilizados após o deadline para submissão do projecto. Além de verificar a correcção do output produzido, o ambiente de avaliação **restringe a memória e o tempo de execução** disponíveis. A maior parte dos testes executa o comando `diff` da forma seguinte:

```
diff output result
```

O ficheiro `result` contém o output gerado pelo executável a partir do ficheiro `input`. O ficheiro `output` contém o output esperado. Um programa passa num teste e recebe o valor correspondente, quando o comando `diff` não reporta quaisquer diferenças (i.e., não produz qualquer output). O sistema reporta um valor entre 0 e 170.

A nota obtida na classificação automática poderá sofrer eventuais cortes caso a análise do código demonstre recurso a soluções ajustadas a inputs concretos ou outputs aleatórios/constantes.

Detecção de Cópias

A avaliação dos projectos inclui um procedimento para detecção de cópias. A submissão de um projecto implica um compromisso de que o trabalho foi realizado exclusivamente pelos alunos. A violação deste compromisso ou a tentativa de submeter código que não foi desenvolvido pelo grupo implica a anulação do trabalho, e consequente classificação de 0, para todos os alunos envolvidos (incluindo os alunos que disponibilizaram o código). Qualquer tentativa de fraude, directa or indirecta, será comunicada ao Conselho Pedagógico do IST e ao coordenador do curso, e será penalizada de acordo com as regras aprovadas pela Universidade e publicadas em “Diário da República”.