INSTITUTO SUPERIOR TÉCNICO

Análise e Síntese de Algoritmos 2024/2025

1° Projecto

Data enunciado: 14 de Novembro de 2024
Data Limite de Entrega: 6 de Dezembro de 2024

Descrição do Problema

O arqueólogo João Caracol encontrou no Templo Perdido da civilização TuttiFrutti várias tabelas que definem operadores binários sobre inteiros como o operador $\oplus$ cuja tabela se apresenta em baixo:

$\oplus$	1	2	3
1	2	2	1
2	3	2	1
3	1	3	3

Mais tarde, numa câmara escondida do templo, o arqueólogo encontrou sequências de aplicações sucessivas dos vários operadores associadas ao respectivo resultado, como aquela que se apresenta em baixo:

$$2 \oplus 2 \oplus 2 \oplus 2 \oplus 1 \oplus 3 = 1$$

No entanto, em nenhuma das sequências figuram os parêntesis que determinam a ordem pela qual as operações devem ser aplicadas. O arqueólogo pretende agora descobrir a ordem pela qual efectuar cada uma das sequências de operações de modo a obter os resultados que lhe estão associados. Por exemplo, uma das ordenações possíveis para a sequência apresentada em cima corresponde à seguinte parentização:

$$((((2 \oplus 2) \oplus 2) \oplus (2 \oplus 1)) \oplus 3) = 1$$

O objectivo do projecto é desenvolver um algoritmo para automatizar a colocação de parêntesis nas sequências encontradas pelo Arqueólogo Caracol.

Input

O ficheiro de entrada contém toda a informação acerca da operação binária a considerar bem como a respectiva sequência de inteiros e resultado associado. Assim, o ficheiro de entrada é definido da seguinte forma:

- Uma linha contendo um inteiro $n \geq 1$ correspondendo ao tamanho do conjunto de inteiros sobre os quais a operação $\oplus$ está definida e um número $m \geq 1$ correspondendo ao tamanho da sequência de inteiros.
- Uma lista de n linhas, em que cada linha i contém n inteiros, cada um com um valor entre 1 e n, representando cada um deles o resultado da operação $\oplus$ aplicado ao inteiro i e ao índice da respectiva coluna. Isto é, o inteiro m na posição j da linha i corresponde ao valor $i \oplus j$.
- Uma linha com uma sequência de m inteiros entre 1 e n separados por espaço, que corresponde à sequência de inteiros aos quais foi aplicada a operação $\oplus$. Os elementos da sequência enontram-se separados por um único espaço em branco, não contendo qualquer outro carácter, a não ser o fim de linha.
- Uma linha com o inteiro correspondente ao resultado desejado.

Output

O programa deverá examinar a cadeia de inteiros dada na penúltima linha do input e decidir se é possível parentetizar a cadeia por forma a que o valor da expressão resultante seja igual ao valor desejado dado na última linha do input. O output deve ser o seguinte:

- Caso seja possível parentizar a sequência de inteiros por forma a obter o valor desejado: o programa deve escrever 1 no `stdout` seguido de quebra de linha e da parentização mais à esquerda.
- Em caso contrário, o programa deve escrever simplesmente 0 no `stdout`.

Exemplo 1

Input

```
2 4
1 2
2 1
1 2 2 1
1
```

Output

```
1
(((1 2) 2) 1)
```

Exemplo 2

Input

```
3 6
3 2 1
3 2 1
1 3 3
2 2 2 2 1 3
1
```

Output

```
1
((((2 2) 2) (2 1)) 3)
```

Implementação

A implementação do projecto deverá ser feita preferencialmente usando a linguagen de programação 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. Mais se observa que soluções recursivas podem esgotar o limite da pilha quando executadas sobre os testes de maior tamanho, pelo que se recomenda a implementação de algoritmos **iterativos**.
O tempo necessário para implementar este projecto é inferior a 15 horas.

Parâmetros de compilação:

```
C++: g++ -std=c++11 -O3 -Wall file.cpp -lm
C: gcc -O3 -ansi -Wall file.c -lm
Javac: javac File.java
Java: java -Xss32m -Xmx256m -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: deve ser submetido através do sistema Mooshak e o relatório (em formato PDF) deverá ser submetido através do Fénix. 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 e para o Fénix. Note que apenas a última submissão será considerada para efeitos de avaliação. Todas as submissões anteriores serão ignoradas: tal inclui o código fonte e o relatório.

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 mémoria 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 reprovação na unidade curricular, para todos os alunos envolvidos (includindo os alunos que disponibilizaram o código). Qualquer tentativa de fraude, directa or indirecta, será comunicada ao Conselho Pedagógico do IST, ao coordenador de curso, e será penalizada de acordo com as regras aprovadas pela Universidade e publicadas em "Diário da República".