

Trabalho Prático 2: Ordenação de Experimentos

Introdução

Neste trabalho, você foi convidado a participar da organização de um grande projeto científico. Um grupo de cientistas definiu um conjunto de experimentos e a relação de dependência entre eles. Seu objetivo é fornecer uma ordem de execução para esses experimentos que respeite essa relação de dependência.

Entrada

Seu programa receberá um arquivo de entrada com uma tabela contendo:

- O número de experimentos.
- O número de dependências entre os pares de experimentos.
- Um conjunto de relações de dependências. A relação de dependência é definida da seguinte forma: **exp2** > **exp1** significa que o experimento **exp2** só poderá ser realizado quando o experimento **exp1** estiver terminado.

A seguir um exemplo de arquivo de entrada:

```
6 5
2 > 3
2 > 4
4 > 1
1 > 0
0 > 5
```

Na primeira linha do arquivo, 6 é o número de experimentos, 5 é o número de dependências entre os pares de experimentos. As linhas seguintes são as relações de dependências entre os experimentos: o experimento 2 só poderá ser realizado depois do experimento 3 e do experimento 4, o experimento 4 só poderá ser realizado depois do experimento 1, e assim por diante.

Saída

O seu programa retornará como resultado uma possível ordenação dos experimentos. Dependendo da entrada, diversas ordenações são possíveis. Neste caso, considere aquela que segue uma ordem crescente dos experimentos, ou seja, experimentos que possuem identificador menor precedem experimentos com identificador maior. Por exemplo, considerando o arquivo de entrada dado, duas ordenações são possíveis: 3 5 0 1 4 2 ou 5 3 0 1 4 2. No entanto, a saída fornecida pelo programa será: 3 5 0 1 4 2. O resultado deverá ser salvo em um arquivo, numa única linha, com os experimentos separados por um espaço, isto é:

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

O que deve ser implementado

Você deve implementar um programa que recebe como parâmetros o nome do arquivo de entrada, contendo os experimentos a serem realizados, e o nome do arquivo de saída onde será impresso a ordenação dos experimentos, no formato descrito anteriormente.

O seu programa deve ser executado da seguinte maneira:

```
./programa entrada.txt saida.txt
```

O que deve ser entregue

- Deve ser submetido o código fonte, seus arquivos `.h` (com as declarações das funções) e seus arquivos `.c` (com a implementação das funções). O programa deve ser ANSI, ou seja, o código fonte do programa deve ser compilável no Linux, não devendo ser utilizada nenhuma biblioteca que seja específica do sistema operacional MS-Windows.
- Documentação sobre o trabalho (em `.pdf`), incluindo as decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado. Um exemplo de documentação pode ser encontrado no Moodle. Porém, ela deve conter, entre outras coisas:
 1. Introdução: descrição sucinta do problema a ser resolvido e visão geral sobre o funcionamento do programa.
 2. Implementação: descrição sobre a implementação do programa. Deve ser detalhada a estrutura de dados utilizada (de preferência com diagramas ilustrativos), o funcionamento das principais funções e procedimentos utilizados, o formato de entrada e saída de dados, compilador utilizado, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado.
 3. Estudo de Complexidade: estudo da complexidade do tempo de execução dos procedimentos implementados e do programa como um todo (notação O), considerando o número de experimentos E e o número de dependências D .
 4. Testes: descrição dos testes realizados e listagem da saída (não edite os resultados).

5. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
 6. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso.
- O trabalho será entregue através do sistema de submissão de trabalhos práticos: <http://aeds.dcc.ufmg> em um arquivo compactado (.zip) contendo os arquivos fonte (arquivo .c, o arquivo .h, o main.c). A documentação deve ser enviada em pdf, no link especial para sua submissão no sistema.

Ponto extra

Suponha que você possua mais recursos para realizar os experimentos e gostaria de paralelizar seu trabalho. Nesse caso, considere que os experimentos poderão ser realizados em duas máquinas. Seu objetivo é fornecer uma ordenação que respeite as relações de dependência mas que explore também a possibilidade dos experimentos serem realizados ao mesmo tempo. Um exemplo de saída para a entrada mostrada anteriormente:

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

Nesse resultado, os experimentos 5 e 3 podem ser realizados em paralelo. Considere que todos os experimentos levam a mesma unidade de tempo para serem realizados.