Trabalho Prático 2

Algoritmos I

Entrega: 20/10/2020

1 Introdução

Adriana é uma ourives muito famosa e conhecida por suas excentricidades. Extremamente metódica, sempre que recebe um malote com diamantes realiza o mesmo procedimento antes da criação de suas joia. Ela faz o chamado **Jogo dos Diamantes** para cada malote.

O jogo dos diamantes assume que para uma coleção de pedras, duas pedras quaisquer podem ser "combinadas". Suponha que tenhamos duas pedras com pesos **p1** e **p2**, com **p1** >= **p2**, o resultado dessa operação pode ser:

- Se **p1** == **p2**, as duas pedras são completamente destruídas.
- Se **p1** > **p2**, a pedra **p1** sobra com um novo peso **p1-p2**, e a pedra **p2** é completamente destruída.

O jogo termina quando resta no conjunto uma, ou nenhuma pedra. E o objetivo é que o peso restante seja o menor possível (caso não reste nenhuma pedra, o peso restante é zero).

2 Definições e Objetivos Específicos

A entrada do problema consiste em um conjunto de diamantes representados por seu respectivo peso em gramas, ou seja, um inteiro positivo por diamante presente no conjunto.

No final desse processo teremos no máximo um diamante. Retorne o peso do diamante restante ou zero (0) caso não sobre nenhum. **Note:** A escolha dos pares de diamantes deve ser ótima, ou seja, o valor retornado no final deve ser o menor peso possível dentre todas as combinações.

Considere um conjunto com no máximo 256 diamantes e peso máximo 128.

3 Exemplo do desafio do diamante

Nessa seção iremos prover, passo a passo, um exemplo do Jogo do Diamante

Considere o conjunto de 6 diamantes com os respectivos pesos: 1, 4, 8, 1, 2, 7.

- Passo 1: Compare as pedras com peso 2 e 4, retorne 2 ao conjunto. (resultando em 1, 2, 8, 1, 7).
- Passo 2: Compare as pedras com peso 7 e 8, retorne 1 ao conjunto. (resultando em 1, 2, 1, 1).
- Passo 3: Compare as pedras com peso 2 e 1, retorne 1 ao conjunto. (resultando em 1, 1, 1).
- Passo 3: Compare as pedras com peso 1 e 1, não retorne nenhuma pedra ao conjunto. (resultando em 1).
- Passo 4: Retorne 1 como solução ótima do problema.

4 Os arquivos de entrada e saída

Exemplo das linhas da entrada

```
K // quantidade de diamantes
D1 D2 .. DK // pesos dos diamantes
```

Exemplo da saída

```
X // Peso retornado.
```

5 Exemplo Prático de Entrada e Saída

Exemplo prático da entrada

```
6
1 4 8 1 2 7
```

Exemplo prático da saída

```
1
```

6 Avaliação Experimental

Para a avaliação experimental o aluno deverá criar entradas para o problema de forma a avaliar se o tempo de execução do algoritmo está de acordo com a complexidade reportada na documentação, aumentando a quantidade de diamantes, guardando os tempos de execução e colocando esses valores em gráficos para melhor visualização.

É necessário executar mais de uma vez o algoritmo para entradas com o mesmo tamanho de entrada, guardando os tempos de execução e reportando a média e o desvio padrão destes tempos. Faça isso para vários valores diferentes de tamanho de entrada, para visualizar bem como o tempo está variando com a variação do tamanho da entrada.

Compare os resultados entre uma solução que use programação dinâmica, e para uma solução que não use. Veja a diferença de tempo de execução, como está aumentando o tempo com o aumento da entrada, e explique a diferença nesses valores.

ATENÇÃO: Todas as regras a seguir deverão ser rigorosamente seguidas. Submissões fora do padrão descrito serão devidamente **penalizadas**.

7 O que deve ser entregue

Você deve submeter um arquivo compacto (zip ou tar.gz) no formato **seu_nome_sua_matrícula** via Moodle contendo:

- todos os arquivos do código *.c*, *.cpp* e *.h* que foram implementados,
- um arquivo *makefile*[1] **que crie um executável tp2**,
  - **ATENÇÃO:** O makefile é para garantir que o código está sendo compilado corretamente, de acordo com o modo que vocês modelaram o programa. É **essencial** que ao digitar "make" na linha de comando dentro da pasta onde reside o arquivo makefile, o mesmo compile o programa e gere um executável chamado **tp2**.
- sua documentação.

Sua documentação deve ter até 10 páginas contendo:

- uma breve descrição do problema,
- explicações das estruturas de dados e dos algoritmos utilizados para resolver o problema. Para tal, artifícios como pseudo-códigos, exemplos ou diagramas podem ser úteis. Note que documentar uma solução não é o mesmo que documentar seu código. Não inclua textos de códigos em sua documentação.
- análise de complexidade da solução proposta (espaço e tempo). Cada complexidade apresentada deverá ser devidamente justificada para que seja aceita.
- avaliação experimental.
- conclusão.

O seu TP deverá ser entregue de acordo com a data especificada no moodle. A **penalidade** em porcentagem para os TPs atrasados é dada pela fórmula $2^{d-1}/0.16$.

[1] https://pt.wikibooks.org/wiki/Programar_em_C/Makefiles

8 Implementação

8.1 Linguagem, Ambiente e Parâmetros

O seu programa deverá ser implementado na linguagem **C** ou **C++** e poderá fazer uso de funções da biblioteca padrão da linguagem. Trabalhos que utilizem qualquer outra linguagem de programação e/ou que façam uso de bibliotecas que não a padrão não serão aceitos. Usaremos o ambiente Linux (mica.dcc.ufmg.br) para rodar os códigos. Note que esse ambiente é acessível a todos os alunos do DCC com seu login e senha. **ATENÇÃO:** Não é necessário que o aluno implemente em ambiente Linux.

ATENÇÃO: Teste seu código na máquina previamente especificada. **DICA:** Aproveite esse momento para conferir a funcionalidade, makefile e demais características do código.

ATENÇÃO: O arquivo da entrada deve ser passado como parâmetros para o programa através da linha de comando (e.g., *$ ./tp2 jogo_do_diamante.txt*) e imprimir a saída no **stdout** (com printf/cout), não em um arquivo.

8.2 Testes

A sua implementação passará por um processo de correção automática, o formato da saída de seu programa deverá ser idêntico aquele descrito nas seções anteriores. Saídas diferentes serão consideradas erro para o programa. É seu dever certificar que seu programa atenda corretamente para qualquer entrada válida para o problema.

8.3 Qualidade do código

É importante prestar atenção para a qualidade do código, mantendo-o organizado e comentado para não surgir dúvidas na hora da correção. Qualquer decisão que não estiver clara dada a documentação e a organização do código será descontada na nota final.

9 Consideração Final

Assim como em todos os trabalhos dessa disciplina é estritamente proibido a cópia parcial ou integra de códigos, seja da internet ou de colegas. Seja honesto! Você não aprende nada copiando código de terceiros. Se a cópia for detectada, sua nota será zerada e o professor será informado para que as devidas providências sejam tomadas.