

Universidade Federal de Minas Gerais

Departamento de Ciência da Computação

Algoritmos I

Trabalho Prático 2

Jogo dos Diamantes

20 de outubro de 2020

1 Introdução

O presente projeto é uma resposta ao problema do Jogo dos Diamantes.

O Jogo dos Diamantes tem por entrada um conjunto de diamantes, sendo cada um identificado simplesmente por seu peso. Um jogador deve, a cada rodada, escolher dois diamantes, e combiná-los. Ao combinar um diamante, existem 2 situações possíveis: os pesos são diferentes, e um restolho de diamante cujo peso é a diferença dos originais é adicionado à lista em lugar destes; ou então os pesos são iguais, e os dois diamantes simplesmente são removidos da lista, sem nada ser colocado de volta.

2 Implementação

Em primeiro lugar, foi implementado um algoritmo de força bruta (classe `brute_force`), que varre todas as possibilidades de escolha de dois diamantes, desde a configuração inicial, e guarda os pesos finais em um vetor (função `find_finals`), para que o mínimo possa ser extraído. Ele se mostrou péssimo no tratamento do problema, tanto que com uma entrada de 10 diamantes, já encontrou dificuldade em responder (alguns minutos), e, para entradas pouco maiores, simplesmente não respondia.

Esse comportamento lentíssimo (talvez mais até do que se esperaria de um algoritmo de complexidade exponencial qualquer) se deve ao fato de que cada possibilidade é calculada e armazenada em um vetor, só sendo descartada depois que todos os problemas filhos foram resolvidos. Assim, rapidamente ocorre também um overload de espaço, forçando o sistema operacional a realizar operações custosas de acesso a níveis mais baixos da hierarquia de memória.

Então, escolheu-se o paradigma de programação dinâmica. Com esse método, chegou-se a um algoritmo eficiente (classe `dynamic`), cujos passos alto-níveis são descrito abaixo.

1. Ordenar os pesos crescentemente;
2. Alocar diamantes em "baldes" de acordo com o peso (um balde por peso)
3. Percorrer o vetor de baldes de trás para frente, combinando sempre um diamante do balde mais à direita, e outro do penúltimo.
4. Parar quando só sobrar um balde. Se o número de diamantes no balde for par, retornar 0. Se não, retornar o peso dos diamantes daquele balde.

A observação que nos leva a esse algoritmo é que o Jogo do Diamante se reduz em remover partes de diamantes grandes e se. A solução que sempre procura arrancar pedaços o maior possível de diamantes o mais pesado possível é ótima.

A subestrutura ótima (de menor peso possível) é expandida dos diamantes mais pesados para os menos pesados durante sua execução.

Segue abaixo um diagrama das classes implementadas no projeto (Figura 1).

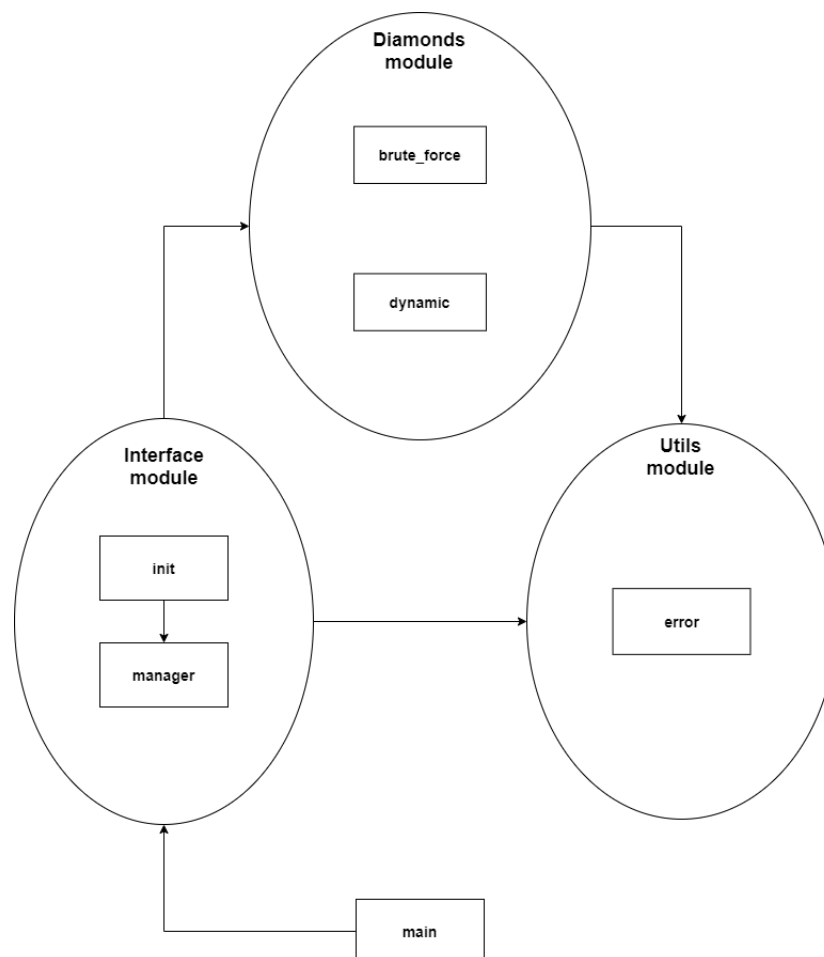


Figura 1: Diagrama de classes

3 Análise de Complexidade

O algoritmo força-bruta claramente é assintoticamente exponencial. Mas e quanto ao de programação dinâmica? Esperaria que fosse quasi-linear, pois a cada iteração pelo menos um diamante some, sendo que o ganho é acumulativo por conta do método utilizado.

Sob inspeção empírica, ele demonstrou ser muito poderoso, tolerando limites muito mais extensos para os parâmetros do que foi imposto inicialmente (número de diamantes ≤ 256 , e peso ≤ 128), como se pode ver no gráfico (Figura 2) - foram utilizados até 500000 diamantes, mas o peso foi mantido em 128, e o número de diamantes foi alterado em passos de 10000, sendo que o código foi executado 10 vezes para cada número de diamantes.

O peso não foi posto sob este tipo de análise, porque é obvio que sua influência sobre a complexidade é negligente, na média (escolhendo valores aleatórios entre zero e o máximo).

O gráfico foi gerado através dos arquivos "run-tests.sh", "src/times-measure.cpp" e "src/times-measure.ipynb".

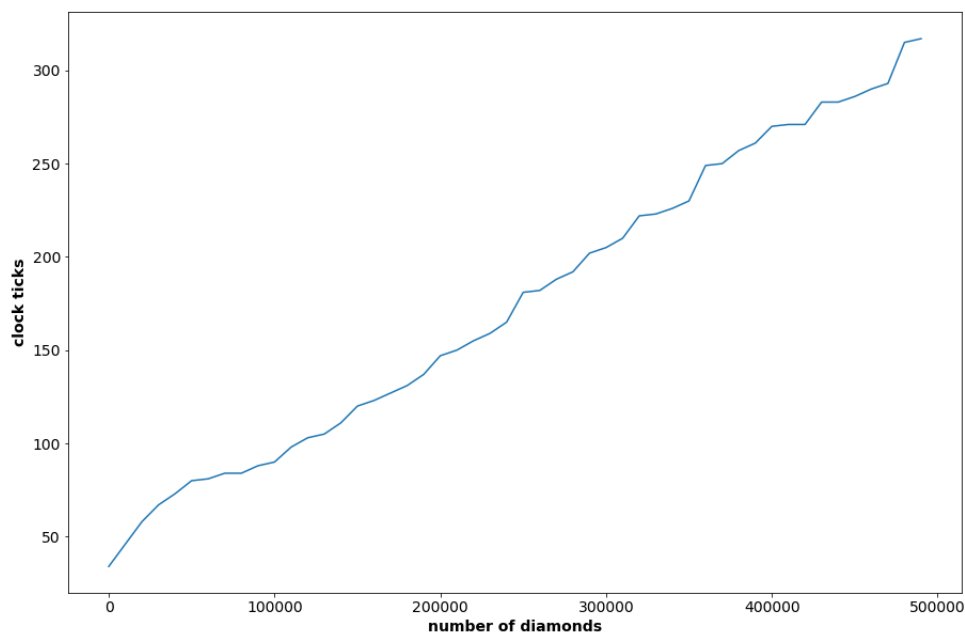


Figura 2: Complexidade temporal

Quanto à complexidade espacial, fica claro que, como inicialmente se tem um contêiner contendo os n pesos, sendo logo após alocado o espaço dos baldes que são no máximo n , é linear, isto é, $O(n)$.

4 Instruções de Compilação e Execução

Para compilar o programa, basta digitar "make" no diretório principal, e deverá ser criado um executável "tp2.exe". Existe também as opções "make clean", para limpar o diretório de executáveis, e "make test", que realiza os procedimentos necessários para gerar arquivo .csv contendo os valores temporais para os testes realizados.

5 Conclusão

Depois da implementação desse projeto, se tornou mais claro o abismo de diferença entre um algoritmo força bruta e outro, que utiliza do paradigma de programação dinâmica. O implacável progresso em direção à solução ótima deste último o torna uma poderosa ferramenta, quando usado na situação adequada.

6 Bibliografia

<http://www.cplusplus.com/reference/>
<https://matplotlib.org/>
<https://ss64.com/>
<https://stackoverflow.com/>