

Trabalho de Programação — Pesquisa Completa & Dividir e Conquistar

Análise e Complexidade de Algoritmos — PPComp

Prof. Dr. Jefferson O. Andrade

Campus Serra — Ifes

2020/2

1 Introdução

Este trabalho visa exercitar, através da aplicação prática em programação, os conceitos de *pesquisa completa* – ou *pesquisa por força bruta* – e *dividir e conquistar*. O trabalho consiste em resolver uma lista de problemas no site Online Judge.¹ Os problemas em questão foram especificamente selecionados, por necessitarem de algoritmos que envolvam as técnicas de programação descrita acima.

Para realização do trabalho será necessário se cadastrar no site Online Judge. O cadastro no site é gratuito e não são exigidas informações pessoais para o cadastros, exceto o endereço de e-mail.

A lista de problemas que devem ser resolvidos encontra-se na Seção 2. Você deve resolver o problema, submetê-lo ao Online Judge. Na situação mais desejável, seu programa deverá obter aprovação, i.e., veredito “Accepted”.

Ao entregar o trabalho, para cada problema você deve entregar um “print” da página de submissões do Online Judge mostrando o veredito que seu programa recebeu. Caso, seu programa não consiga o veredito “Accepted”, você ainda pode enviá-lo, mas ele receberá uma nota menor.

Caso você esteja tendo dificuldades com algum dos problemas, um recurso que pode ajudá-lo é o uDebug. Este site contém exemplos de entradas e das respectivas saídas esperadas para cada problema. Se o seu programa não receber veredito “Accepted”, mas conseguir resolver os testes do uDebug corretamente, ele será considerado parcialmente correto.

2 Problemas

Os problemas estão divididos em seções, de acordo com a técnica que deve ser utilizada para resolvê-los. Pode ser que outras estratégias sejam utilizadas em conjunto, mas a estratégia indicada deve estar presente e identificável.

2.1 Pesquisa Completa

Nesta seção estão os problemas que podem ser solucionados por pesquisa completa. Podendo envolver tanto busca sequencial iterativa, i.e., *laços*, quanto pesquisa do espaço de soluções em árvore por largura ou por profundidade com ou sem *backtracking*, i.e., *recursão*.

¹O site era conhecido anteriormente como “UVa Online Judge”, onde “UVa” é a sigla da Universidad de Valladolid.

1. 10908 – Largest Square
2. 00628 – Passwords
3. 00598 – Bundling Newspaper

2.2 Dividir e Conquistar

Os problema nesta seção devem ser solucionados utilizando algoritmos baseados na estratégia de dividir e conquistar.

1. 10706 – Number Sequence
2. 11935 – Through the Desert

3 Sobre a Entrega do Trabalho

O trabalho deve ser entregue, através do sistema AVA institucional do Ifes. A data e horário limites para a entrega estarão definidas na atividade do AVA.

Para cada problema deve ser entregue o código fonte do programa que resolve o problema, um arquivo de imagem com o veredito dado pelo Online Judge, e um arquivo `Readme.md`, em formato Markdown, contendo:

- Nome do autor.
- Uma breve explicação de como a técnica indicada foi usada no programa.
- A análise da complexidade de tempo do programa desenvolvido.
- Outras informações que o autor julgar apropriadas para o entendimento do trabalho realizado.

O arquivo `Readme.md` será visualizado pelo professor utilizando a extensão *Markdown Viewer* para o navegador Google Chrome.

Os arquivos de cada problema devem ser gravados em um diretório nomeado segundo o código numérico do problema indicado. Por exemplo, a solução do problema “00105 – The Skyline Problem”² deveria estar organizada da seguinte forma:

```
00105
├── 00105.cpp
├── 00105-veredito.png
└── Readme.md
```

O trabalho deve ser entregue **obrigatoriamente** em um arquivo no formato ZIP contendo todos os itens descritos acima.

Atenção: o professor fará o download de todos os arquivos de resposta do AVA para um diretório local e há um *script de shell* irá descompactar os arquivos, testar os programa e gerar um relatório preliminar. Se o arquivo não estiver no formato indicado ou com estrutura de diretórios diferente da especificada, o *script* irá indicar programa com erro! Esse é o comportamento desejado do *script*.

²Este problema não faz parte do trabalho, ele está sendo usado apenas como exemplo.

4 Sobre a Execução do Trabalho

- Os problemas podem ser solucionados em qualquer uma das linguagens aceitas pelo Online Judge (C, C++, Java, Pascal e Python). Preste atenção nas versões das linguagens aceitas pelo Online Judge.
- Os problemas não precisam ser todos resolvidos na mesma linguagem.
- O trabalho deve ser desenvolvido **individualmente**.

5 Sobre a Avaliação do Trabalho

Serão levados em consideração na correção do trabalho os seguintes fatores:

- Corretude do resultado final.
- Organização do código.
- Clareza do código.
- Modularidade do código.
- Conformidade com o enunciado do trabalho.

Para este trabalho, devido ao seu escopo reduzido, o principal componente da avaliação será a corretude do resultado final. Serão considerados três níveis de corretude:

1. Programa com veredito “Accepted” do Online Judge.
2. Programa que resolve **corretamente** os testes do uDebug.
3. Programa que resolve **parcialmente** os testes do uDebug.
4. Programa incorreto.

A pontuação de cada problema será dada, em parte, pelo nível de correção.

Importante: Uma vez que os problemas do *Online Judge* são públicos é quase certo que haja soluções disponíveis para estes problemas na Internet. Não copie soluções da Internet (ou de colegas) sob hipótese alguma! Um sistema para detecção de similaridade de software será utilizado para avaliar todas as repostas enviadas. Caso sejam detectadas cópias, todas as instâncias de cópia receberão nota zero.