

Trabalho II — Projeto e Desenvolvimento de Algoritmos

Teoria da Computação — PPComp

Prof. Dr. Jefferson O. Andrade

Campus Serra — Ifes

2021/2

1 Introdução

Este trabalho visa exercitar, através da aplicação prática em programação, os conceitos de projeto, desenvolvimento e análise de complexidade de algoritmos. O trabalho consiste em resolver uma lista de problemas no site Online Judge.¹

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. Não é necessário receber o veredito “Accepted” para enviar o seu programa. 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 ajudar 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 que devem ser resolvidos estão indicados abaixo. Eles estão organizados em categorias de acordo com a técnica sugerida para a solução do problema.

Há 10 problemas no total. Você deve escolher 8 para resolver. O professor **não irá corrigir** mais do que 8 problemas por entrega. Se você entregar mais de 8 problemas o professor irá executar uma função randômica para determinar quais dos problemas que você entregou serão corrigidos.

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

- Problemas Ad Hoc
 1. 12247 – Jollo
 2. 00447 – Population Explosion
- Pesquisa Completa
 1. 10487 – Closest Sums
 2. 00167 – The Sultan Successor
- Dividir e Conquistar
 1. 11935 – Through the Desert
 2. 11621 – Small Factors
- Algoritmos Gulosos
 1. 11269 – Setting Problems
 2. 10602 – Editor Nottobad
- Programação Dinâmica
 1. 00983 – Localized Summing for ...
 2. 11368 – Nested Dolls

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.
- 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

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

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*.

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.
- Análise de complexidade do programa

O principal componente da avaliação será a corretude do resultado final. Serão considerados quatro níveis de corretude:

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

O segundo elemento mais importante na avaliação do trabalho será a análise de complexidade. Entre 20% e 40% da pontuação de cada questão verá da análise de complexidade.

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.