

# Trabalho de Programação – Programação Dinâmica & Algoritmos Gulosos

Tópicos de Programação Avançada – BSI

Prof. Dr. Jefferson O. Andrade

Campus Serra – Ifes

2021/1

## 1 Introdução

Este trabalho visa exercitar, através da aplicação prática em programação, os conceitos de *dividir e conquistar*, *programação dinâmica* e *algoritmos gulosos* estudados na disciplina de Tópicos de Programação Avançada.

O trabalho consiste em resolver uma lista de problemas no site Online Judge. Os problemas em questão foram especificamente selecionados, por necessitarem de algoritmos que envolvam programação dinâmica.

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 cadastro, 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 e obter a 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.

**Atenção:** todos os programas enviados serão submetidos a uma *ferramenta de verificação de plágio*, então procure não copiar códigos da internet ou de seus colegas.

## 2 Problemas

1. 12190 – Electric Bill

2. 10131 – Is Bigger Smarter?
3. 00562 – Dividing Coins
4. 10670 – Work Reduction
5. 10020 – Minimal coverage

### 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 de programação dinâmica foi usada no programa.
- Uma descrição de como compilar (se for o caso) e executar o programa.
- Outras informações que o autor julgar apropriadas para o entendimento do trabalho realizado.

Um exemplo do conteúdo do arquivo README.md é dados abaixo:

```
# Trabalho de Programação
**Dividir e Conquistar & Programação Dinâmica & Algoritmos Gulosos**

**Autor:** Jefferson O. Andrade

**Data:** 25 de agosto de 2020

**Problema:** 00105 -- The Skyline Problem

## Sobre a Solução

Este diretório contém o código fonte gerado para solucionar o problema 00105 do *Online Judge*. O problema recebeu veredito "Accepted", como mostrado na figura abaixo:

![Veredito](./00105-veredito.png)

O programa foi desenvolvido em C++11. Para compilar o programa deve ser usado o seguinte comando:

...
g++ -std=c++11 00105.cpp -o 00105
...
```

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”<sup>1</sup> deveria estar organizada da seguinte forma:

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

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

## 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.

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 incorreto.

---

<sup>1</sup>Este problema não faz parte do trabalho, ele está sendo usado apenas como exemplo.