

# Problemas sobre Estruturas de Dados Básicas

## Tópicos de Programação Avançada

Prof. Dr. Jefferson O. Andrade

2019/2

## Contents

1	Introdução	1
2	Lista de Problemas	1
3	Execução do Trabalho	2
4	Entrega	3

## 1 Introdução

Este trabalho consiste na resolução de um conjunto de problemas envolvendo as estruturas de dados básicas vistas (ou revistas) na disciplina de *Tópicos Avançados de Programação*.

Todos os problemas propostos neste trabalho estão disponíveis no site UVa Online Judge. Portanto será necessário que ao menos um dos integrantes do grupo autor do trabalho faça cadastro no site. O cadastro no site é gratuito e não são exigidas informações pessoais para o cadastros, exceto o endereço de e-mail.

O conjunto de problemas que devem ser resolvidos pelo grupo está listado na Seção 2, agrupados por tópicos.

## 2 Lista de Problemas

### 1. *Warm up*

- UVa 12247 – Jollo

### 2. Vetores (Arrays 1D – C++ STL vector; ou Java Vector/ArrayList)

- UVa 10038 – Jolly Jumpers
- UVa 11340 – Newspaper

### 3. Matrizes (Arrays 2D)

- UVa 10920 – Spiral Tap
- UVa 11581 – Grid successors

4. Ordenação (C++ STL algorithm; ou Java Collections)
  - UVa 10107 – What is the Median?
  - UVa 10258 – Contest Scoreboard
5. Manipulação de bits (C++ STL bitset; Java BitSet; bitmask)
  - UVa 10264 – The Most Potent Corner
  - UVa 11926 – Multitasking
6. Lista Encadeada (C++ STL list; ou Java LinkedList)
  - UVa 11988 – Broken Keyboard...
7. Pilhas (C++ STL stack; Java Stack)
  - UVa 00514 – Rails
  - UVa 01062 – Containers
8. Filas (C++ STL queue/deque; Java Queue/Deque)
  - UVa 10172 – The Lonesome Cargo
  - UVa 10901 – Ferry Loading III
9. Árvore Binária de Pesquisa (balanceada) (C++ STL map; Java TreeMap)
  - UVa 00939 – Genes
  - UVa 10132 – File Fragmentation
10. Conjuntos (C++ STL set; Java TreeSet)
  - UVa 00978 – Lemmings Battle
  - UVa 11849 – CD

### 3 Execução do Trabalho

Este trabalho pode ser executado por grupos de até 3 alunos.

Os problemas do trabalho podem ser implementados em qualquer uma das linguagens aceitas pelo *UVa Online Judge*. Devem ser utilizadas apenas as bibliotecas padrão de cada linguagem, caso contrário, o programa não executará no *UVa Online Judge*.

Uma vez que o grupo tenha codificado a solução para o problema, esta solução deve ser submetida ao juiz online. Idealmente, as soluções entregues neste trabalho devem ter sido aceitas pelo juiz online. O grupo deve incluir uma imagem comprovando que a solução foi aceita.

Caso o grupo não consiga aprovação da solução por extrapolação do limite de tempo (*Time Limit Exceeded – TLE*), mas o grupo tenha convicção de que a solução está correta, a solução ainda poderá ser entregue acompanhada de um texto indicando que o grupo testou a solução para diferentes entradas e obteve resultados corretos (preferencialmente usando o uDebug).

## 4 Entrega

O grupo deve entregar em um arquivo em formato ZIP os seguintes artefatos:

- Todos os códigos fontes gerados para solução dos problemas.
- Relatório de execução do trabalho em  $\text{\LaTeX}$ .

O código fonte para solução de cada problema deverá estar gravado em um diretório com o nome igual ao código do problema – i.e., `uva12247`, `uva10038`, `uva11340`, etc. Caso a solução do grupo não tenha recebido aceite do juiz online, mas resolva corretamente o problema, o grupo deve incluir no diretório arquivos de exemplo que possam ser usados para testar o programa.

Todos os arquivos fontes **sem exceção** devem conter um comentário no início do arquivo indicando que problema aquele programa resolve, e quem são os autores.

O relatório de execução do trabalho deve ser escrito em  $\text{\LaTeX}$ , e deve ter, além da introdução, uma seção para cada estrutura de dados. Em cada seção devem constar as soluções dos problemas relacionados àquela estrutura de dados. A solução deve ter uma introdução breve sobre como o programa funciona, seguida da listagem do programa. Para mostrar a listagem do programa utilize o pacote `minted` do  $\text{\LaTeX}$ , e faça a inclusão do arquivo com o código fonte. Mostre as linhas das listagens de código numeradas.

A apresentação do relatório pode valer até 10% da nota final do trabalho.