

Universidade Tecnológica Federal do Paraná (UTFPR)  
Departamento Acadêmico de Informática (DAINF)  
Prof: Rodrigo Minetto (rodrigo.minetto@gmail.com)  
Estrutura de dados I  
Trabalho Prático

## 1 Objetivo

O objetivo desse trabalho prático é aprofundar a compreensão em alguns dos tópicos visto no curso: tipo abstrato de dados Fila, Pilha, Lista, Back-Tracking, Ordenação e/ou Busca. Você **deve escolher uma aplicação ou problema** usando algum(ns) destes conceitos. Obs: trabalhos triviais, ou seja, que considerem simplesmente operações (inserção, remoção, seleção...) nestas estruturas receberão nota zero.

Exemplos de aplicações ou problemas interessantes: backtracking (que usa uma pilha de chamadas recursivas) para resolução de algum problema interessante, filas de prioridade para escalonamento de processos em um sistema operacional, jogos ou problemas que utilizam algum dos conceitos estudados, algoritmos mais avançados que utilizam as estruturas de dados que estudamos em sua composição, resolução de expressão numéricas com pilha, parser de linguagens de programação (verificar se por exemplo no html se um tag fecha com outro), resolução de problemas de xadrez, ...

## 2 Composição do trabalho

O trabalho, realizado individualmente, é composto pelas partes:

- Código-fonte (zip ou tar contendo somente Makefile, .c e .h).
- **Relatório** (pdf), com a descrição da aplicação ou problema, e como o tópico escolhido aborda o(s) conceito(s) visto(s) na disciplina. É necessário uma descrição dos testes, bem como soluções encontradas **OU** um **vídeo** (que pode ser postado no youtube) com no máximo 7 minutos e que contenha uma apresentação do trabalho (descrição do problema, motivação, etc) com execuções de alguns testes para mostrar o funcionamento do programa.

Modularize o que for possível no seu código, implemente funções razoavelmente curtas, separe a parte de definição das operações (tipo abstrato de dados) da parte de implementação (.c e .h). Crie um **Makefile** para compilar o seu código. Seu código deve ser comentado e indentado. Ao executar o Makefile, devem ser gerados executáveis para cada um dos testes.

Obs1: a falta de Makefile (funcionando) e/ou a presença de qualquer código relacionado a code-Bocks (e IDEs ...) implicarão na não correção do seu trabalho (todo o trabalho).

Obs2: a apresentação do trabalho não está descartada, e pode ser usada para identificar alunos que não realizaram o trabalho.

## 3 Informações sobre a entrega

- **Data de entrega no moodle** do relatório (ou vídeo no youtube ou enviado em formato avi dentro do ZIP) e código (em arquivo único compactado): 22/Agosto/2021 (domingo) as 23h59.

## 4 Informações adicionais

- Cópias (com ou sem eventuais disfarces) receberão nota ZERO.
- Trabalho atrasados serão descontados em 50% da nota por dia de atraso.
- O relatório e os códigos devem ser entregues via **Moodle** (se preferir o vídeo pode ser postado no youtube, basta passar o link na submissão em um README). Identifique o trabalho com o registros acadêmicos do aluno (sem espaços): XXXX.{zip,tgz} (só esses formatos de compactação).
- A nota do trabalho será composta por: 40% referente a implementação e 60% referentes ao relatório (trabalhos sem código funcionando invalidam o relatório).
- Relatórios devem ter capa, descrição do problema, solução, testes, ... A ausência de cada um desses itens reduz em 25% a nota do relatório.
- As notas serão proporcionais a dificuldade do problema, ou seja, problemas mais complexos e soluções mais elaboradas receberão notas mais altas que trabalhos com problemas mais simples.