

## Projetos TDA para Estruturas de Dados

Os grupos deverão implementar um Tipo Abstrato de Dados na forma de um arquivo `.h` e um arquivo de implementação (arquivo `.c`) como mostrado em aula. Espera-se que os alunos pesquisem o tema sob sua responsabilidade e dominem tanto a estrutura de dados como as rotinas utilizadas para implementar as operações. Poderá ser utilizado código de terceiros (livros, apostilas, *sites*, etc) como ponto de partida, mas é imprescindível que o grupo conheça a fundo todo o código utilizado no seu TDA. Testar bem as implementações antes de entregar no dia 8 de junho de 2016. Após a entrega o professor apresentará a cada grupo um conjunto de problemas em *sites online judges* para cada tema para serem resolvidos com a implementação. A nota levará em conta a qualidade da implementação, se a equipe conseguiu solucionar os problemas de *online judges* propostos e o grau de conhecimento de cada membro da equipe sobre o TDA apresentado.

Os grupos de alunos são listados na próxima página e, em princípio, poderão sugerir o tema que pretendem trabalhar. Se mais de um grupo se interessar pelo mesmo tema, será realizado um sorteio para decidir quem ficará com o tema em questão. Na aula teórica de 16 de março de 2016 definiremos os temas das equipes, lembrando que todos os membros de uma equipe devem ser do mesmo grupo de laboratório e que cada equipe deve ter 3 elementos.

Os temas propostos estão descritos nas páginas a seguir.

## Projetos TDA para Estruturas de Dados

- **Números romanos**

TDA para representar números romanos e suportar operações básicas.

Estruturas de dados para representar:

- Número romano, na forma de uma string

Operações a suportar:

- Verificar se uma string é um número romano válido
- Converter para maiúsculas
- Converter para minúsculas
- Converter de romano para decimal
- Converter de decimal para romano
- Operações aritméticas básicas: adição, subtração, multiplicação, divisão
- Comparação entre dois números romanos

Referências básicas:

- [http://en.wikipedia.org/wiki/Roman\\_numerals](http://en.wikipedia.org/wiki/Roman_numerals)
- [http://www.novaroma.org/via\\_romana/numbers.html](http://www.novaroma.org/via_romana/numbers.html)
- [http://pt.wikipedia.org/wiki/Numera%C3%A7%C3%A3o\\_romana](http://pt.wikipedia.org/wiki/Numera%C3%A7%C3%A3o_romana)
- <http://www.somatematica.com.br/tabela1.php>
- <http://numeracaoromana.babuo.com/numeros-romanos-de-1-a-5000>
- <http://math.hws.edu/eck/cs124/javanotes3/c9/ex-9-3-answer.html>
- <http://stackoverflow.com/questions/17479287/convert-int-to-roman-numeral>
- <http://www.sanfoundry.com/c-program-convert-numbers-roman/>

## Projetos TDA para Estruturas de Dados

- **Grafos**

TDA para representar grafos com pesos e sem pesos e suportar operações básicas.

Estruturas de dados para representar:

- Grafo por matriz de adjacências
- Grafo por lista de adjacências (por contiguidade)
- Indicar a qtde de vértices
- Indicar a qtde de arestas

Operações a suportar:

- Determinar a árvore geradora mínima
- Determinar o menor caminho (Dijkstra) entre dois vértices
- Percorrer o grafo em largura
- Percorrer o grafo em profundidade

Referências básicas:

- Capítulos 9 e 10 do livro *Programming Challenges* de Revilla & Skiena, disponível na pasta “Maratona” da área pública da rede
- Material do curso de programação com grafos na pasta “Maratona” da área pública da rede
- Ver o material disponível na pasta `CursoMaratonaUFOP` disponível na pasta “Maratona” da área pública da rede

## Projetos TDA para Estruturas de Dados

- **Strings**

TDA para representar textos de até 64Kb e operações elementares básicas.

Estruturas de dados para representar:

- String de até 64Kb

Operações a suportar:

- Receber frase
  - Ler com `fgets` e normalizar o final da string, descartando caracteres de controle do final
  - Deixar pré computado o comprimento da string
- Converter string para maiúsculas
- Converter string para minúsculas
- Determinar quantas palavras tem a string
- Remover espaços em branco excedentes
  - Do início, do final e do meio da string
- Determinar substring
- Implementar KMP ou outro algoritmo eficiente de pesquisa de substrings

Referências básicas:

- Capítulo 3 do livro *Programming Challenges* de Revilla & Skiena, disponível na pasta “Maratona” da área pública da rede
- Ver arquivo `UFMG-TRD-2011.pdf` disponível na pasta “Maratona” da área pública da rede
- Ver o material disponível na pasta `CursoMaratonaUFOP` disponível na pasta “Maratona” da área pública da rede
- Ver livro do Cormen, Leiserson, Rivest e Stein na biblioteca da Fatec, que fala do KMP

## Projetos TDA para Estruturas de Dados

- **Lista circular**

TDA para manipular lista circular, ou seja, lista em que após o elemento final da fila considera-se o como seguinte o elemento inicial da mesma.

Estruturas de dados para representar:

- Uma lista contendo um número indeterminado de inteiros, que poderá ser percorrida nos dois sentidos de direção (a partir do início e a partir do final).
- Um descritor para a lista

Operações a suportar:

- Criar e inicializar a lista.
- Inserir elementos na lista
- Excluir elementos da lista
- Imprimir o conteúdo da lista
- Pesquisar um elemento na lista

Referências básicas:

- TENENBAUM, Aaron M; LANGSAM, Yedidiah; AUGENSTEIN, Moshe J. ESTRUTURAS DE DADOS USANDO C. 1ª ed. São Paulo: Makron Books (Pearson Education). 1995 (tem na biblioteca da Fatec e exemplifica inclusive com o problema de Josephus)
- CELES, W.; CERQUEIRA, R.; RANGEL, J. L. Introdução a Estruturas de Dados: com técnicas de programação em C. Rio de Janeiro: Campus, 2004. (tem em pdf na pasta do Sergião, de ED)
- VELOSO, Paulo et alli. Estrutura de dados. 4ª ed. Rio de Janeiro: Campus, 1986 (tem na biblioteca da Fatec)
- Praticamente qualquer livro sobre ED trata desse assunto
- Ver arquivo UFMG-TRD-2011.pdf disponível na pasta “Maratona” da área pública da rede

## Projetos TDA para Estruturas de Dados

- **Sudoku**

TDA para representar uma instância do problema padrão do Sudoku e operações elementares.

Estruturas de dados para representar:

- Arranjo composto por 3x3 quadrados, onde cada quadrado é composto por 9 células.

Operações a suportar:

- Criar o arranjo
- Verificar se o conteúdo do arranjo resolve o problema
- Gerar um problema de Sudoku aleatório que seja válido para ser resolvido
- Solucionar um problema válido de Sudoku
- Imprimir o arranjo

Referências básicas:

- <http://www.mathblog.dk/project-euler-96-su-doku/>

## Projetos TDA para Estruturas de Dados

- **Datas**

TDA para representar datas no formato convencional gregoriano e suas operações básicas.

Estruturas de dados para representar:

- Dia, mês e ano.

Operações a suportar:

- Validação de datas
- Ano é bissexto?
- Aritmética de datas
  - Quantos dias existem entre duas datas informadas?
  - Quantos meses existem entre duas datas informadas?
  - Quantos anos existem entre duas datas informadas?
  - Se somarmos uma quantidade de dias a uma data, que nova data teremos?
- Dia da semana correspondente a uma data válida informada

Referências básicas:

- Rotina de cálculo de dia da semana vista na disciplina de Linguagens de Programação
- <http://www.vivaolinux.com.br/dica/Calculo-de-dias-uteis-entre-duas-datas-em-PHP> (em PHP, pode ser interessante para bolar problemas e para ter ideias de operações sobre datas)
- [http://www.planetariodorio.com.br/index.php?option=com\\_k2&view=item&id=294:c%C3%A1lculo-dos-dias-da-semana](http://www.planetariodorio.com.br/index.php?option=com_k2&view=item&id=294:c%C3%A1lculo-dos-dias-da-semana) (tem um macete sobre dias julianos que pode ser interessante para calcular dias corridos)
- <https://sites.google.com/site/sergioeducacaofinanceira/artigos-publicados/feriados> (serve para cálculo da Páscoa?)
- [http://petitcomputer.wikia.com/wiki/How\\_to\\_get\\_the\\_number\\_of\\_days\\_between\\_dates](http://petitcomputer.wikia.com/wiki/How_to_get_the_number_of_days_between_dates) (este aqui parece que dá certo para dias corridos, precisa testar bem e ver a questão das divisões, se são todas inteiras ou não)
- [http://www.science20.com/news\\_articles/how\\_date\\_easter\\_calculated\\_science\\_answer-78344](http://www.science20.com/news_articles/how_date_easter_calculated_science_answer-78344) (acho que dá para calcular a data da Páscoa e, com isso, todos os feriados móveis)

## Projetos TDA para Estruturas de Dados

- **Conjuntos**

TDA para representar conjuntos de inteiros e suportar operações básicas de manipulação de conjuntos.

Estruturas de dados para representar:

- Até 10000 inteiros de 32 bits

Operações a suportar:

- Pertinência de elemento a conjunto
- Equivalência de conjuntos
- Cardinalidade do conjunto
- Operações clássicas: União, Intersecção, Diferença, Produto cartesiano
- Operações básicas: Criação, Destruição, Inclusão, Exclusão, Impressão

Referências básicas:

- Exercício específico da primeira lista de ED
- Procurar o tema Union-Find e implementar o algoritmo



## Projetos TDA para Estruturas de Dados

- **Geometria plana**

TDA elementar para representação de elementos de geometria em duas dimensões, com suporte a algumas operações básicas.

Estruturas de dados para representar:

- Pontos no plano
- Linhas no plano
- Polígonos planos

Operações geométricas a suportar:

- Determinar a distância entre dois pontos
- Determinar o comprimento de uma linha
- Verificar se um polígono é válido a partir de suas medidas
- Determinar a área do triângulo e do quadrilátero
- Verificar se três pontos são colineares
- Verificar onde duas retas se interceptam

Referências básicas:

- Capítulo 14 do livro *Programming Challenges* de Revilla & Skiena, disponível na pasta “Maratona” da área pública da rede
- <http://www.dpi.inpe.br/gilberto/livro/geocomp/geometria.pdf>
- Ver arquivo `library.pdf` disponível na pasta “Maratona” da área pública da rede
- Ver arquivo `icpc.pdf` disponível na pasta “Maratona” da área pública da rede
- Ver arquivo `UFMG-TRD-2011.pdf` disponível na pasta “Maratona” da área pública da rede
- Ver o material disponível na pasta `CursoMaratonaUFOP` disponível na pasta “Maratona” da área pública da rede

## Projetos TDA para Estruturas de Dados

- **Polinômios**

TDA para representar polinômios de uma variável e suportar operações básicas sobre tais estruturas.

Estruturas de dados para representar:

- Lista de termos que compõem o polinômio

Operações a suportar:

- Operações básicas: Criação, Destruição, Inclusão, Exclusão, Impressão
- Normalização/simplificação
  - Agrupar todos os termos de mesmo grau em um único termo
  - Colocar os termos em ordem decrescente de grau
- Determinar o grau do polinômio
- Valor do polinômio para um dado X informado

Referências básicas:

- Exercício específico da primeira lista de exercícios de ED

## Projetos TDA para Estruturas de Dados

- **Posfixo/Infixo/Prefixo**

TDA para manipular e solucionar expressões aritméticas expressas nas formas pós-fixa, infixa e pré-fixa.

Estruturas de dados para representar:

- Uma expressão aritmética válida, que pode conter dígitos, os operadores das 4 operações aritméticas básicas e parênteses.

Operações a suportar:

- Criar e armazenar uma expressão, indicando se é pré, pós ou infixa.
- Converter a expressão de uma representação para outra
- Determinar o resultado de uma expressão válida

Referências básicas:

- <http://www.mathblog.dk/tools/infix-postfix-converter/>
- TENENBAUM, Aaron M; LANGSAM, Yedidyah; AUGENSTEIN, Moshe J. ESTRUTURAS DE DADOS USANDO C. 1ª ed. São Paulo: Makron Books (Pearson Education). 1995 (tem na biblioteca da Fatec)
- <https://www.mcs.uvawise.edu/dar5p/prevcourses/2010012180/presentations/class13.pdf>
- Ver arquivo UFMG-TRD-2011.pdf disponível na pasta “Maratona” da área pública da rede
- Ver arquivo estrut-dados-pucrio.pdf, pp 122-125, disponível na área pública da rede.

## Projetos TDA para Estruturas de Dados

- **Deque**

TDA para manipular um deque (*double ended queue*), ou seja, uma fila em que as inclusões e exclusões podem ocorrer tanto no seu início como no seu final.

Estruturas de dados para representar:

- Uma lista contendo um número indeterminado de inteiros, que poderá ser percorrida nos dois sentidos de direção (a partir do início e a partir do final).
- Um descritor para a lista

Operações a suportar:

- Criar e inicializar a lista.
- Inserir elementos na lista
- Excluir elementos da lista
- Imprimir o conteúdo da lista
- Pesquisar um elemento na lista

Referências básicas:

- Livros de ED em geral