



Estruturas de Dados I

Plano de Ensino

Turma B
Prof. Marcelo Cezar Pinto
1º semestre de 2014

| | | |
|----------|---|----------|
| 1 | Identificação | 1 |
| 2 | Objetivo geral do Curso | 2 |
| 3 | Ementa | 2 |
| 4 | Objetivos | 2 |
| 4.1 | Geral | 2 |
| 4.2 | Específicos | 2 |
| 5 | Cronograma e conteúdos programáticos | 2 |
| 6 | Procedimentos metodológicos | 3 |
| 6.1 | Os encontros acadêmicos em sala de aula | 3 |
| 6.2 | Os encontros acadêmicos em laboratório | 4 |
| 6.3 | A observância da frequência | 4 |
| 6.4 | As listas de exercícios | 4 |
| 6.5 | O fórum | 4 |
| 6.6 | As Minimaratonas | 4 |
| 7 | Avaliação do processo ensino-aprendizagem | 6 |
| 7.1 | A composição da primeira nota parcial | 6 |
| 7.2 | A composição da segunda nota parcial | 6 |
| 7.3 | As Avaliações Escritas | 6 |
| 7.4 | O Trabalho | 6 |
| 7.5 | A penalidade por plágio ou fraude | 7 |
| 8 | Recuperação: novas oportunidades de aprendizagem e avaliação | 7 |
| 9 | Bibliografia | 7 |
| 9.1 | Bibliografia básica | 7 |
| 9.2 | Bibliografia complementar | 7 |
| 9.3 | Bibliografia sugerida | 7 |

1 Identificação

Curso Ciência da Computação

Créditos 4

Turno Noturno

Carga horária (em horas) 60

Componente curricular Estruturas de Dados I

Carga horária (em horas-aula) 72

Turma B

Professor

Fase 2ª

Prof. Marcelo Cezar Pinto

Atendimento ao estudante*

segundas e quartas-feiras das 17:00 às 18:30

*Outros horários poderão ser agendados através do fórum.

2 Objetivo geral do Curso

O curso tem por objetivo a formação integral de novos cientistas e profissionais da computação, os quais deverão possuir conhecimentos técnicos e científicos e serem capazes de aplicar estes conhecimentos, de forma inovadora e transformadora, nas diferentes áreas de conhecimento da Computação. Adicionalmente, os egressos do curso deverão ser capazes de adaptar-se às constantes mudanças tecnológicas e sociais, e ter uma formação ao mesmo tempo cidadã, interdisciplinar e profissional.

3 Ementa

¹Alocação dinâmica de memória. ²Variáveis estáticas e dinâmicas. ³Listas lineares e suas generalizações: listas ordenadas, listas encadeadas, pilhas e filas. ⁴Complexidade de algoritmos. ⁵Algoritmos de busca e ordenação. ⁶Implementações com linguagem imperativa estruturada.

- Numeramos os itens da ementa a fim de melhor esclarecermos como o cronograma (Seção 5) deste Plano contempla cada item.

4 Objetivos

4.1 Geral

Utilizar estruturas de dados básicas e avançadas para a solução de problemas computacionais e analisar as soluções propostas para determinar suas complexidades.

4.2 Específicos

- Aprimorar os conhecimentos fundamentais sobre algoritmos necessários à disciplina de Estruturas de Dados, bem como a relação profunda entre algoritmos e recursividade.
- Implementar na prática os principais algoritmos e estruturas estudados, especialmente num sistema de proporções suficientes a exigir as competências de organização, modularização e elegância de código.
- Amadurecer as habilidades de programação tendo em foco a corretude e a complexidade das soluções computacionais.
- Compreender as estruturas de dados clássicas e sua importância para a Ciência da Computação.
- Dominar os algoritmos clássicos de organização de dados, especialmente os algoritmos de ordenação e busca.
- Evoluir as capacidades acadêmicas de escrita, defesa e argumentação formal sobre os resultados produzidos.
- Desenvolver as habilidades necessárias para competições de programação.

5 Cronograma e conteúdos programáticos

| Data | Programa |
|--------------|--|
| Março | |
| 17 | Apresentação do Plano de Ensino e Introdução a competições de Programação ⁶ |
| 19 | Revisão de algoritmos clássicos ⁴ |
| 24 | 1ª Minimaratona ⁶ |
| 26 | Introdução à Notação Assintótica e à Complexidade de Algoritmos ⁴ |
| 31 | 2ª Minimaratona ⁶ |

| Data | Programa |
|--------------|---|
| Abril | |
| 02 | Discussão sobre os métodos de avaliação e estratégias de aprendizado. ⁶ |
| 07 | Recursão ⁵⁶ |
| 09 | Recursão ⁵⁶ |
| 14 | 3ª Minimaratona ⁶ |
| 16 | Busca Binária ⁴⁵⁶ |
| 21 | Feriado de Tiradentes |
| 23 | 4ª Minimaratona ⁶ |
| 28 | A função <code>qsort</code> da <code>stdlib.h</code> e Exemplos de problemas de Maratona envolvendo ordenação ⁵⁶ |
| 30 | 5ª Minimaratona ⁶ |
| 05 | 1ª Avaliação Escrita ⁴⁵⁶ |
| Maiο | |
| 07 | Resolução da 1ª Avaliação Escrita e Vista da NP_1 ⁴⁵⁶ |
| 12 | Recuperação da NP_1 ⁴⁵⁶ |
| 14 | Alocação Dinâmica de Memória, Revisão de Ponteiros e Tipos Abstratos de Dados ¹²⁶ |
| 19 | Tipos Abstratos de Dados ¹²⁶ |
| 21 | Especificações do Trabalho ¹²³⁴⁶ |
| 26 | Listas encadeadas ¹²³⁶ |
| 28 | Listas duplamente encadeadas e circulares ¹²³⁶ |
| Junho | |
| 02 | Filas ¹²³⁶ |
| 04 | Pilhas ¹²³⁶ |
| 09 | Desenvolvimento do Trabalho em Laboratório ¹²³⁴⁶ |
| 11 | Exercícios em Laboratório diversos sobre Listas, Filas e Pilhas ¹²³⁶ |
| 16 | Métodos quadráticos de ordenação ⁴⁵⁶ |
| 18 | Análise dos métodos quadráticos de ordenação ⁴⁵⁶ |
| 23 | <i>Merge Sort</i> ⁴⁵⁶ |
| 25 | Análise do <i>Merge Sort</i> ⁴⁵⁶ |
| 30 | <i>Quick Sort</i> ⁴⁵⁶ |
| Julho | |
| 02 | Análise do <i>Quick Sort</i> e <i>Quick Sort</i> Aleatorizado ⁴⁵⁶ |
| 07 | Exercícios em Laboratório: Métodos de ordenação em Listas Encadeadas ¹²³⁴⁶ |
| 09 | 2ª Avaliação Escrita ¹²³⁴⁵⁶ |
| 14 | Resolução da 2ª Avaliação Escrita ¹²³⁴⁵⁶ |
| 16 | Apresentação do Trabalho e Vista da NP_2 ¹²³⁴⁵⁶ |
| 21 | Recuperação da NP_2 ¹²³⁴⁵⁶ |

6 Procedimentos metodológicos

6.1 Os encontros acadêmicos em sala de aula

Nos encontros acadêmicos que ocorrerem em sala de aula, os professores unirão as turmas e ministrarão as aulas de modo expositivo-constructivista, explorando a participação dos estudantes e desenvolvendo com eles exemplos práticos com implementação e compilação em tempo real. Nestes encontros, utilizar-se-ão a lousa, a lousa interativa, o projetor e eventuais recursos didáticos complementares.

6.2 Os encontros acadêmicos em laboratório

Nos encontros acadêmicos que ocorrerem nos laboratórios, os estudantes terão a oportunidade de eles mesmos resolverem problemas computacionais e implementarem suas soluções, quer seja nas Minimaratonas que serão realizadas (cf. Seção 6.6), quer seja nas oportunidades que terão para desenvolverem o Trabalho (cf. Seção 7.4). Nestes encontros, utilizar-se-ão os laboratórios 403 e 404 (campus Chapecó, bloco B) do Curso de Ciência da Computação, com todos os recursos dos quais dispõem.

6.3 A observância da frequência

A frequência do estudante nas aulas será observada através de listas de presença. O estudante deverá assinar a lista de presença no começo de um encontro para se reconhecer como presente na primeira aula e reassinar a lista no final do encontro para se reconhecer como presente na última aula.

Em conformidade com o que estabelecem o Ministério da Educação e Cultura e a Universidade Federal da Fronteira Sul, para ser aprovado, o estudante precisa ter no mínimo 75% de frequência nos encontros presenciais, o equivalente a 54 horas-aula neste caso. Portanto, o estudante que tiver mais de 18 faltas estará automaticamente reprovado, sem direito a recuperação, independentemente de suas notas.

Os professores só abonarão faltas mediante instrução oficial da Coordenação do Curso.

6.4 As listas de exercícios

Serão publicadas no *Moodle* diversas listas de exercícios. O estudante poderá postar as suas resoluções no fórum ou apresentá-las aos professores no horário de atendimento, para discutir os conteúdos com os professores e com os demais colegas. Embora não seja atribuída nota para as listas de exercícios, adverte-se fortemente que *a resolução de todos os exercícios é o principal meio através do qual o estudante pode adquirir as habilidades necessárias para obter um bom desempenho nas avaliações.*

6.5 O fórum

A página do curso no *Moodle* dispõe de um fórum, no qual os professores publicarão eventuais avisos às turmas e no qual os estudantes poderão postar dúvidas, organizar grupos de estudos, agendar horários de atendimento extra, entregar resoluções de exercícios e debater sobre quaisquer outros tópicos pertinentes.

6.6 As Minimaratonas

Visando atingir os objetivos declarados na Seção 4.2, serão promovidas 5 Minimaratonas de Programação, todas com 2 problemas e 90 minutos de competição, nos moldes da Maratona de Programação da Sociedade Brasileira de Computação, com duas diferenças:

1. Enquanto que na Maratona da SBC os estudantes competem em time de 3 integrantes, nas Minimaratonas de Estruturas de Dados os estudantes deverão competir individualmente.
2. Na Maratona da SBC, os competidores podem consultar materiais impressos ou manuscritos. Nas Minimaratonas de Estruturas de Dados, cada estudante poderá levar uma mídia removível (*pen-drive*, *CD* ou *DVD*) com seu material de consulta.

As Minimaratonas ocorrerão nos laboratórios do Curso e começarão sempre 10 minutos após o início do encontro acadêmico para o qual foram agendadas (cf. Seção 5). O regulamento das competições, discriminado a seguir, será discutido com os estudantes no primeiro encontro acadêmico do semestre letivo e constará na contracapa dos cadernos de problemas que cada estudante receberá antes do início das provas.

• Sobre a competição

1. A competição tem duração de 90 minutos.

2. Durante, e apenas durante, o tempo de competição, o estudante deverá tentar resolver os 2 problemas presentes no caderno de problemas que lhe foi entregue antes do início da competição, submetendo o código-fonte de suas resoluções conforme explicado adiante.
3. É proibida qualquer comunicação entre os competidores, bem como consultas a qualquer material além daquele presente na mídia removível trazido pelo próprio estudante.
4. O total de *balões* faturados nesta competição será adicionado ao *saldo de balões* do estudante para compor a nota M da NP_1 .

• **Sobre a entrada**

1. A entrada de seu programa deve ser lida da *entrada padrão*.
2. A entrada é composta por um único caso de teste, descrito em um número de linhas que depende do problema.
3. Quando uma linha da entrada contém vários valores, estes são separados por um único espaço em branco; a entrada não contém nenhum outro espaço em branco.
4. Cada linha, incluindo a última, contém o caractere final-de-linha.
5. O final da entrada coincide com o final do arquivo.

• **Sobre a saída**

1. A saída de seu programa deve ser escrita na saída padrão.
2. Quando uma linha da saída contém vários valores, estes devem ser separados por único espaço em branco; a saída não deve conter nenhum outro espaço em branco.
3. Cada linha, incluindo a última, deve conter o caractere final-de-linha.

• **Sobre o código-fonte**

1. Seu programa deve ser escrito em C ou C++, submetido respectivamente com a extensão `.c` ou `.cpp`, e compilável respectivamente com:

```
gcc -Wall -lm
```

ou

```
g++ -Wall -lm
```

2. Você deve submeter o código-fonte nomeado exatamente como indicado no enunciado do problema.

• **Sobre os pareceres de avaliação**

NO -- FILENAME MISMATCH O código submetido não foi nomeado apropriadamente.

NO -- COMPILATION ERROR Ocorreu algum erro de compilação.

NO -- TIME LIMIT EXCEEDED A execução do programa para algum caso de teste não parou dentro do limite de tempo esperado.

NO -- RUNTIME ERROR Ocorreu algum erro durante a execução do programa para algum caso de teste, como falha de segmentação ou exceção de ponto flutuante.

NO -- WRONG ANSWER A saída impressa pelo seu programa para algum caso de teste não confere com a saída esperada.

NO -- PRESENTATION ERROR A saída impressa pelo seu programa para algum caso de teste não confere com a saída esperada apenas na quantia de espaços em branco.

YES Seu código foi aceito e você faturou um balão! Parabéns!

O total de balões B faturados pelo estudante em todas as 5 Minimaratonas comporá a nota M (Equação 3) pela seguinte Equação:

$$M = \frac{20B}{\max B + 10} \quad (1)$$

sendo $\max B$ o maior saldo de balões dentre todos os estudantes, considerando as duas turmas juntas.

7 Avaliação do processo ensino–aprendizagem

Todas as notas descritas nesta Seção serão consideradas no intervalo $[0,0;10,0]$ com precisão de uma só casa decimal, valendo sempre o arredondamento para cima.

7.1 A composição da primeira nota parcial

A primeira nota parcial é dada por:

$$NP_1 = \max\left\{NP_1', \frac{NP_1' + NP_1^R}{2}\right\} \quad (2)$$

sendo:

$$NP_1' = 0,4P_1 + 0,6M \quad (3)$$

e sendo as notas NP_1^R , P_1 e M como descritas respectivamente nas Seções 8, 7.3 e 6.6.

7.2 A composição da segunda nota parcial

A segunda nota parcial é dada por:

$$NP_2 = \max\left\{NP_2', \frac{NP_2' + NP_2^R}{2}\right\} \quad (4)$$

sendo:

$$NP_2' = 0,4P_2 + 0,6T \quad (5)$$

e sendo as notas NP_2^R , P_2 e T como descritas respectivamente nas Seções 8, 7.3 e 7.4.

7.3 As Avaliações Escritas

Os professores aplicarão duas Avaliações Escritas, ambas individuais e sem consulta. A nota da primeira, denotada por P_1 na Equação 3, figura na composição da primeira nota parcial, e a nota da segunda, denotada por P_2 na Equação 5, na composição da segunda nota parcial. As datas em que as Avaliações ocorrerão se podem conferir na Seção 5.

As Avaliações ocorrerão nos laboratórios do Curso de Ciência da Computação e terão início dez minutos após o início do encontro e duração de 90 minutos. Na ocasião, o estudante receberá uma folha de questões e uma folha para rascunhos, as quais poderá levar para casa. As respostas das questões devem ser postadas em formato .txt na respectiva atividade do Moodle. Após, o estudante assinará uma lista de presença, a qual lhe concederá presença em todas as aulas daquele encontro acadêmico e lhe servirá de garantia. As assinaturas dessa lista serão submetidas à análise comparativa com as assinaturas das demais listas de presença, visando comprovar a identidade do estudante.

O estudante que faltar a uma Avaliação deverá solicitar segunda chamada para a Coordenação do Curso. A segunda chamada será aplicada em horário extraclasse, previamente agendado pelos professores.

No primeiro encontro acadêmico após a realização de uma Avaliação Escrita, os professores comentarão as resoluções da questão em sala de aula — não se obrigando, contudo, a publicar os gabaritos — e permitirão aos estudantes reverem suas respostas e revisarem a correção dos professores. Ademais, o estudante também poderá solicitar vista das Avaliações no horário de atendimento ao estudante.

7.4 O Trabalho

A fim de que o estudante implemente na prática os principais algoritmos e estruturas estudados, especialmente num sistema de proporções suficientes a exigir as competências de organização, modularização e elegância de código, será solicitado um Trabalho, cuja nota T contribuirá para a composição da nota NP_2 . Maiores detalhes sobre o Trabalho serão fornecidos nas Especificações, as quais serão divulgadas e discutidas na data prevista na Seção 5.

Alguns encontros acadêmicos serão destinados ao desenvolvimento do Trabalho em laboratório. No entanto, adverte-se que, se os estudantes não dispuserem de tempo extraclasse para conclusão de suas atividades, dificilmente conseguirão uma boa nota no Trabalho.

7.5 A penalidade por plágio ou fraude

Se for detectada a ocorrência de plágio ou fraude em qualquer instrumento de avaliação ou de recuperação, o estudante infrator terá a nota correspondente anulada. Se for julgado necessário, encaminhar-se-á uma denúncia à Coordenação do Curso para que medidas mais severas sejam tomadas.

8 Recuperação: novas oportunidades de aprendizagem e avaliação

Visando recuperar as notas NP_1 e NP_2 , serão aplicadas duas avaliações escritas, com as mesmas características das demais Avaliações Escritas, cujas notas são designadas respectivamente por NP_1^R e NP_2^R nas Equações 2 e 4. Qualquer estudante pode se submeter a uma recuperação, mesmo que sua nota já esteja acima de 6,0. Observe-se que uma recuperação nunca abaixa a nota de um estudante e que recupera não só a nota da respectiva Avaliação Escrita. Os estudantes que não quiserem fazer uma recuperação não levarão falta no respectivo encontro acadêmico (cf. Seção 5) desde que tenham comparecido à Avaliação Escrita anterior. Os conteúdos cobrados na recuperação da NP_1 serão os mesmos cobrados na 1ª Avaliação Escrita, bem como os cobrados na recuperação da NP_2 serão os mesmos cobrados na 2ª Avaliação Escrita.

9 Bibliografia

9.1 Bibliografia básica

1. CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C.
Algoritmos: Teoria e Prática,
Rio de Janeiro: Campus, 2002.
2. EDELWEISS, N.; GALANTE, R.
Estruturas de Dados,
Porto Alegre: Bookman, 2009.
3. ZIVIANI, N.
Projeto de Algoritmos com Implementações Pascal e C,
2ª edição. São Paulo: Thomson Pioneira, 2004.

9.2 Bibliografia complementar

1. CELES, W.; CERQUEIRA, R.; RANGEL, J. L.
Introdução a Estrutura de Dados: uma Introdução com Técnicas de Programação em C.
Rio de Janeiro: Campus, 2004.
2. WIRTH, N.
Algoritmos e Estruturas de Dados.
Rio de Janeiro: LTC, 1989.

9.3 Bibliografia sugerida

1. SEDGEWICK, R.; WAYNE, K.
Algorithms,
4ª edição. Pearson, 2011.
Disponível em: <http://algs4.cs.princeton.edu/home/>
2. DASGUPTA, S.; PAPANIMITRIOU, C. H.; VAZIRANI, U.
Algoritmos,
McGraw Hill, 2008.

3. KNUTH, D. E.
The Art Of Computer Programming, vol. 1–4
Addison-Wesley, 2011.

Prof. Marcelo Cezar Pinto

Prof. Denio Duarte
Coordenador do Curso

Don't blame me for the fact that competent programming, as I view it as an intellectual possibility, will be too difficult for "the average programmer" — you must not fall into the trap of rejecting a surgical technique because it is beyond the capabilities of the barber in his shop around the corner.

Edsger W. Dijkstra