



Tópicos Especiais em Eng. de *Software*: Técnicas Avançadas de Programação

Plano de Ensino

Prof. Leandro M. Zatesko
1º semestre de 2014

1	Identificação	1
2	Objetivo geral do Curso	1
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	A observância da frequência	3
6.2	O fórum	3
6.3	As Atividades	3
7	Avaliação do processo ensino-aprendizagem	4
7.1	A composição da primeira nota parcial	5
7.2	A composição da segunda nota parcial	5
7.3	A penalidade por plágio ou fraude	5
8	Recuperação: novas oportunidades de aprendizagem e avaliação	5
9	Bibliografia	6
9.1	Bibliografia básica	6
9.2	Bibliografia complementar	6

1 Identificação

Curso Ciência da Computação

Créditos 4

Turno Matutino

Carga horária (em horas) 60

Componente curricular Tópicos Especiais em Engenharia de *Software*

Carga horária (em horas-aula) 72

Fase 7ª

Professor

Prof. Leandro Miranda Zatesko

Atendimento ao estudante* segundas-feiras das 14:00 às 17:00

*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

¹Combinatória e resolução de problemas computacionais por equações de recorrência. ²Algoritmos de divisão e conquista. ³Backtracking. ⁴Algoritmos gulosos. ⁵Programação dinâmica. ⁶Branch and bound.

- 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

Estudar técnicas avançadas de programação para a resolução de problemas complexos, praticando a implementação das técnicas em diversos exercícios e comparando analiticamente o resultado com os algoritmos de força bruta.

4.2 Específicos

- Entender o conceito de ‘espaço de busca’ em problemas de otimização.
- Aprimorar a criatividade e as habilidades necessárias para competições de programação.
- Estudar algumas técnicas para a obtenção em tempo viável de soluções aproximadas para problemas \mathcal{NP} -completos ou \mathcal{NP} -difíceis.

5 Cronograma e conteúdos programáticos

Data	Programa
Março	
17	Apresentação do Plano de Ensino
21	Introdução a competições de Programação
24	Algoritmos clássicos e Problemas <i>ad-hoc</i> ¹²
28	Algoritmos clássicos e Problemas <i>ad-hoc</i> ¹²
31	<i>Strings</i> ¹²
Abril	
04	<i>Strings</i> ¹²
07	Ordenação ¹²
11	Ordenação ¹²
14	Aritmética e Álgebra ¹²
18	Paixão de Cristo
21	Tiradentes
25	Aritmética e Álgebra ¹²
28	Combinatória ¹²⁶
Maiο	
02	Combinatória ¹²⁶
05	Teoria dos Números ¹²⁶
09	Teoria dos Números ¹²⁶
12	Algoritmos gulosos ⁴
16	Algoritmos gulosos ⁴
19	Programação Dinâmica ⁵
23	Programação Dinâmica ⁵

Data	Programa
26	Buscas em Grafos ³⁴⁵
30	Buscas em Grafos ³⁴⁵
Junho	
02	Algoritmos sobre Grafos ³⁴⁵
06	Algoritmos sobre Grafos ³⁴⁵
09	<i>Backtracking</i> ³
13	<i>Backtracking</i> ³
16	Geometria ¹²³⁴⁵⁶
20	Geometria ¹²³⁴⁵⁶
23	Geometria Computacional ¹²³⁴⁵
27	Geometria Computacional ¹²³⁴⁵
30	Encerramento do curso

6 Procedimentos metodológicos

6.1 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 encontros com três aulas, o estudante só não será reconhecido como presente na segunda aula se não o for em ambas as outras.

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.

O professor só abonará faltas mediante instrução oficial da Coordenação do Curso.

6.2 O fórum

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

6.3 As Atividades

Serão promovidas 13 Atividades, 6 das quais serão utilizadas na composição da NP_1 e 7 das quais na composição da NP_2 . Cada Atividade consiste de uma lista de problemas, publicada no *Moodle* de acordo com os Calendários abaixo, sempre às 0:00.

Calendário das Atividades 1–6 (NP_1)	
Número da Atividade	Data da publicação
1	24/03/2014
2	31/03/2014
3	07/04/2014
4	14/04/2014
5	28/04/2014
6	05/05/2014

Calendário das Atividades 7–13 (NP_2)	
Número da Atividade	Data da publicação
7	12/05/2014
8	19/05/2014
9	26/05/2014
10	02/06/2014
11	09/06/2014
12	16/06/2014
13	23/06/2014

Uma vez que uma Atividade tenha sido publicada, o estudante terá 168 horas para tentar resolver o maior número de problemas que conseguir. Cada vez que o estudante quiser submeter uma solução para um problema, deverá postar no respectivo tópico do fórum do *Moodle* o código-fonte designando qual problema está tentando resolver. O professor, então, julgará no prazo de 24 horas o código postado com um dos pareceres listados a seguir.

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 programa para algum caso de teste não confere com a saída esperada.

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

YES O código foi aceito.

A partir do momento em que um problema for resolvido por um estudante, submissões posteriores para o mesmo problema serão desconsideradas, ainda que de outros estudantes. Isto não significa, entretanto, que um estudante não possa fazer uma submissão para um problema existindo uma submissão prévia ainda não julgada de outro estudante para o mesmo problema. Submissões de estudantes que já tenham atingido 3 submissões aceitas na respectiva Atividade também não serão consideradas.

Denotando-se por T_{1-6} o total de submissões aceitas de um estudante pelas Atividades 1–6, compõe-se a nota A'_{1-6} pela Equação:

$$A'_{1-6} = \frac{10T_{1-6}}{\max\{9, \max T_{1-6}\}},$$

sendo $\max T_{1-6}$ a maior nota T_{1-6} da turma. De igual modo, denotando-se por T_{7-13} o total de submissões aceitas de um estudante pelas Atividades 7–13, compõe-se a nota A'_{7-13} pela Equação:

$$A'_{7-13} = \frac{10T_{7-13}}{\max\{10,5; \max T_{7-13}\}},$$

sendo $\max T_{7-13}$ a maior nota T_{7-13} da turma.

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:

$$NP_1 = \max\left\{A_{1-6}, \frac{A_{1-6} + A_{1-6}^R}{2}\right\} \quad (1)$$

sendo A_{1-6} a nota correspondente às Atividades 1–6 (Seção 6.3) e A_{1-6}^R a nota correspondente às Atividades de Recuperação 1–6 (Seção 8).

7.2 A composição da segunda nota parcial

A segunda nota parcial é dada por:

$$NP_2 = \max\left\{A_{7-13}, \frac{A_{7-13} + A_{7-13}^R}{2}\right\} \quad (2)$$

sendo A_{7-13} a nota correspondente às Atividades 7–13 (Seção 6.3) e A_{7-13}^R a nota correspondente às Atividades de Recuperação 7–13 (Seção 8).

7.3 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

Os estudantes poderão recuperar suas notas através de 13 Atividades de Recuperação, 6 delas para comporem a NP_1 e 7 delas para comporem a NP_2 . Cada Atividade de Recuperação consiste de uma lista de problemas, publicada no *Moodle* de acordo com o Calendário abaixo, sempre às 0:00.

Calendário das Atividades de Recuperação	
Número da Atividade	Data da publicação
1	01/04/2014
2	08/04/2014
3	15/04/2014
4	29/04/2014
5	06/05/2014
6	13/05/2014
7	20/05/2014
8	27/05/2014
9	03/06/2014
10	10/06/2014
11	17/06/2014
12	24/06/2014
13	01/07/2014

Uma vez que uma Atividade de Recuperação tenha sido publicada, o estudante terá 168 horas para tentar resolver o maior número de problemas que conseguir. Cada vez que o estudante quiser submeter uma solução para um problema, deverá postar no respectivo tópico do fórum do *Moodle* o código-fonte designando qual problema está tentando resolver. O professor, então, julgará no prazo de 24 horas o código postado com um dos pareceres listados na Seção 6.3.

A partir do momento em que um problema for resolvido por um estudante, submissões posteriores para o mesmo problema serão desconsideradas, ainda que de outros estudantes. Isto não significa, entretanto, que um estudante não possa fazer uma submissão para um problema existindo uma submissão prévia ainda não julgada de outro estudante para o mesmo problema. Submissões de estudantes que já tenham atingido 3 submissões aceitas na respectiva Atividade não serão consideradas.

Denotando-se por T_{1-6}^R o total de submissões aceitas de um estudante pelas Atividades de Recuperação 1–6, compõe-se a nota A_{1-6}^R pela Equação:

$$A_{1-6}^R = \frac{10T_{1-6}^R}{\max\{9, \max T_{1-6}^R\}},$$

sendo $\max T_{1-6}^R$ a maior nota T_{1-6}^R da turma. De igual modo, denotando-se por T_{7-13}^R o total de submissões aceitas de um estudante pelas Atividades de Recuperação 7–13, compõe-se a nota A_{7-13}^R pela Equação:

$$A_{7-13}^R = \frac{10T_{7-13}^R}{\max\{10, 5; \max T_{7-13}^R\}},$$

sendo $\max T_{7-13}^R$ a maior nota T_{7-13}^R da turma.

9 Bibliografia

9.1 Bibliografia básica

1. SKIENA, S. S.; REVILLA, M.
Programming Challenges,
1ª edição. Springer, 2003.
2. DASGUPTA, S.; PAPADIMITRIOU, C. H.; VAZIRANI, U.
Algoritmos,
McGraw Hill, 2008.
3. KLEINBERG, J.; TARDOS, E.
Algorithm Design,
1ª edição. Addison Wesley, 2005.
4. SKIENA, S. S.
The Algorithm Design Manual,
2ª edição. Springer, 2010.

9.2 Bibliografia complementar

1. BELLMAN, R.; DREYFUS, S.
Dynamic Programming,
1ª edição. Princeton University Press, 2010.
2. CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C.
Algoritmos: Teoria e Prática,
Rio de Janeiro: Campus, 2002.
3. KNUTH, D. E.
The Art Of Computer Programming, vol. 1–4
Addison-Wesley, 2011.
4. MANBER, U.
Introduction To Algorithms: A Creative Approach,
1ª edição. Addison-Wesley, 1989.