



116297 - Tópicos Avançados em Computadores - Turma x

Plano de ensino

Prof. Guilherme Novaes Ramos

1 Objetivo

A disciplina visa complementar a formação do aluno em prática de solucionamento de problemas, através de treinamento para competições de programação.

2 Ementa

- Algoritmos e estruturas de dados
- Paradigmas de soluções de problemas
- Matemática computacional
- Cadeias

3 Procedimento de ensino

No início do semestre, um nível $n \in \{1, 2, 3, 4\}$ é atribuído a cada aluno, que está de acordo com a experiência do aluno em competições de programação. O nível 1 representa um aluno iniciante e o nível 4 representa um aluno experiente.

- Atividades teóricas: alunos de nível $n > 1$ devem acompanhar alunos de nível $n - 1$, sanando dúvidas e ensinando algoritmos/técnicas/conceitos. Caso não haja alunos de nível $n - 1$, os alunos de nível n devem acompanhar os alunos de nível $n - 2$ e assim por diante.
- Atividades práticas: alunos de nível $n < 4$ devem participar de torneios de programação e completar as listas de exercícios de seus respectivos níveis. Cada lista de exercícios é uma lista de problemas de juízes automáticos a serem resolvidos.
- Elaboração e manutenção de material didático: alunos de nível 4 devem preparar os torneios e manter atualizadas as listas de exercícios.

4 Regras

- Os torneios devem ser feitos individualmente por cada aluno.
- Cada torneio segue o formato da Maratona de Programação (<http://maratona.ime.usp.br/> → “Regras” → “Formato do concurso”), adaptado para acontecer em quatro horas.
- Durante os torneios, é permitido consultar qualquer material manuscrito ou impresso, como livros, cadernos, apostilas, etc. No entanto, *não* é permitido nenhum tipo de comunicação entre alunos, ou entre um aluno e o mundo externo ao ambiente do torneio.
- Os torneios serão abertos para pessoas de fora participarem.

5 Avaliação

- Cada torneio será composto de oito problemas, onde cada nível é contemplado com dois problemas. A nota de um aluno de nível n no i -ésimo torneio é $T_i = 10 \min\{2n, r_i\}/(2n)$, onde r_i é o número de problemas resolvidos por este aluno neste torneio. A nota final de um aluno nos torneios é $T = (T_1 + T_2 + T_3 + T_4)/4$.
- A nota de um aluno de nível n na lista de exercícios é $E = 10e/t_n$, onde e é o número de exercícios da lista do nível n resolvidos por este aluno; e t_n é o número total de exercícios na lista do nível n .
- Seja $k(n)$ o nível pelo qual o nível n é responsável (de acordo com a Seção 3).

Definimos $M(n) = 1/A_{k(n)} \sum_{i=1}^{A_{k(n)}} E_i$, onde $A_{k(n)}$ é o número de alunos no nível $k(n)$ e E_i é a nota do i -ésimo aluno de nível $k(n)$ na lista de exercícios.

- A nota em material didático de um aluno de nível 4 é $D = 10 \min\{\lfloor 32/A_4 \rfloor, d\} / (\lfloor 32/A_4 \rfloor)$, onde A_4 é o número de alunos no nível 4 e d é o total de problemas elaborados por este aluno para os torneios.
- A nota final de um aluno de nível $n < 4$ é $N = (2T + 2E + M(n))/5$. Se não há alunos de nível menor que n , $N = (T + E)/2$.
- A nota final de um aluno de nível 4 é $N = (M(4) + 4D)/5$. Se não há alunos de nível menor que 4, $N = D$.
- A menção de um aluno é dada da seguinte maneira.

$$\text{Menção} = \begin{cases} \text{SR} & \text{se } N \in [0, 0.1) \\ \text{II} & \text{se } N \in [0.1, 3) \\ \text{MI} & \text{se } N \in [3, 5) \\ \text{MM} & \text{se } N \in [5, 7) \\ \text{MS} & \text{se } N \in [7, 9) \\ \text{SS} & \text{se } N \in [9, 10] \end{cases}$$

6 Cronograma

- 09/04: Torneio 1
- 07/05: Torneio 2
- 04/06: Torneio 3
- 09/07: Torneio 4

7 Bibliografia

- Halim, S., & Halim, F. (2013). *Competitive Programming 3: The New Lower Bound of Programming Contests: Handbook for ACM ICPC and IOI Contestants*. Lulu.com.
- Skiena, S. S., & Revilla, M. A. (2006). *Programming challenges: The programming contest training manual*. Springer.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms*. MIT Press.
- Skiena, S. S. (2008). *The algorithm design manual*. Springer.
- Dasgupta, S., Papadimitriou, C. H., & Vazirani, U. (2006). *Algorithms*. McGraw-Hill, Inc.
- Preparata, F. P., & Shamos, M. (2012). *Computational geometry: an introduction*. Springer.
- Diestel, R. (2010). *Graph theory*. Springer.
- Graham, R. L., Knuth D. E., & Patashnik O. (1994). *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley.

Brasília, 12 de março de 2016

Prof. Guilherme Novaes Ramos