

Programação Competitiva

Laboratório de Programação Competitiva I

Pedro Henrique Paiola

Rene Pegoraro

Wilson M Yonezawa

Sobre a disciplina

- Disciplina optativa - 4 créditos
- Foco nos alunos do 2º ano do BCC
 - Não voltada para alunos que ainda não sabem programar
- Objetivo primário
 - Preparação para competições de programação, como a Maratona de Programação
- Instrutores
 - Paiola (coach das equipes da Unesp Bauru), Rene, Wilson e competidores veteranos

Competições de programação

- Competições de Programação são provas com duração pré-determinada onde os participantes são desafiados a resolver uma série de problemas, usando linguagens de programação específicas.
- Os problemas são de entrada e saída textuais, em que o formato de ambas é bem definido.
- Os códigos são julgados automaticamente por um *software* juiz. Basicamente o juiz irá comparar a saída gerada por seu programa com a saída esperada.

Competições de programação

- Possíveis respostas do juiz:
 - AC - Accepted
 - WA - Wrong Answer
 - PE - Presentation Error
 - TLE - Time Limit Exceeded
 - RE - Runtime Error
 - CE - Compilation Error

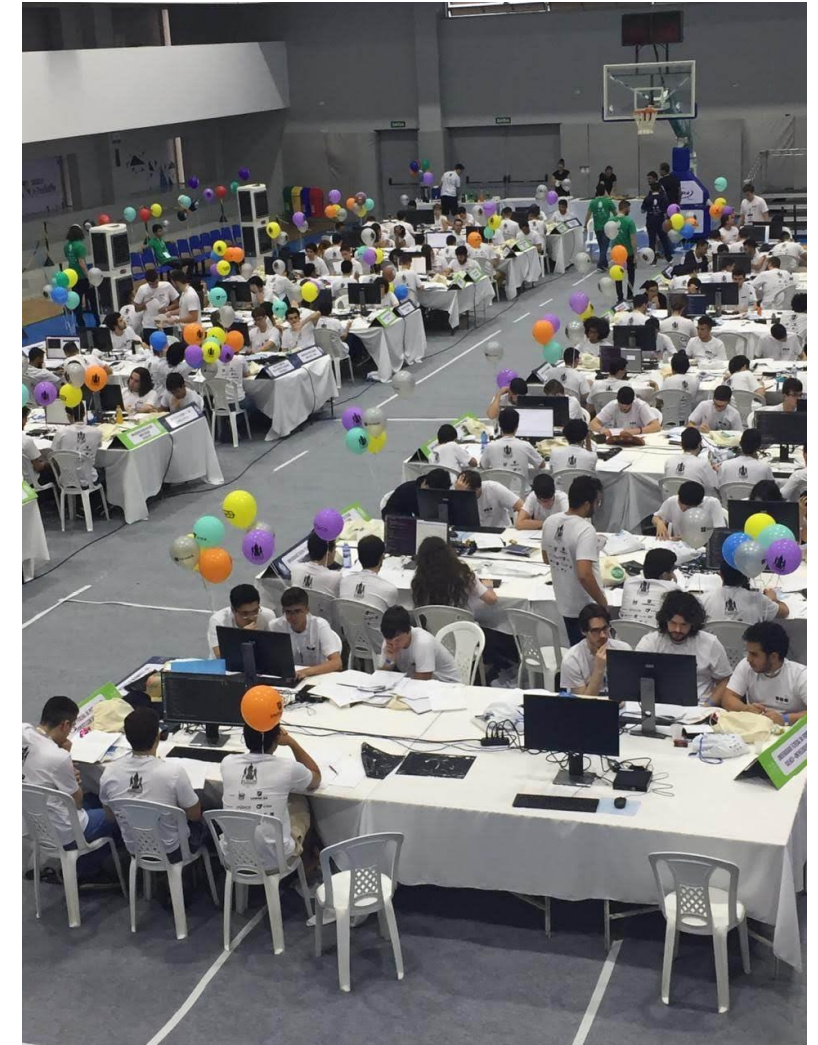
Maratona SBC de Programação

- Evento da Sociedade Brasileira de Computação (SBC), realizado desde 1996.
- Destinada a alunos e alunos de cursos de graduação e início de pós na área de Computação e afins.
- Classificatória para as finais mundiais do concurso de programação, o International Collegiate Programming Contest.



Maratona SBC de Programação

- Ocorre em duas fases:
 - Fase Regional: normalmente em setembro
 - Final Brasileira: normalmente em novembro
- Times de 3 pessoas
- 1 computador por grupo
- Permitido consulta de material impresso
- 5 horas de *contest*
- Em torno de 12 problemas a serem resolvidos



Por quê?

1. Conhecimento e resolução de problemas

- Você é confrontado com diversos problemas, todo muito diferentes, envolvendo diversas técnicas e estruturas de dados específicas.
- A maioria dos problemas vão além do óbvio, requisitando o domínio de diversos conhecimentos e criatividade para combiná-los e aplica-los adequadamente.
- Estudando programação competitiva você irá aprender assuntos, de forma prática e aplicada, que o curso de graduação só irá oferecer mais tarde (ou nem irá oferecer).

Por quê?

2. Trabalho em equipe

- A Maratona de Programação é obrigatoriamente feita em grupos de três pessoas.
- O grupo terá que aprender a lidar com a escassez de recursos e tempo para resolver o maior número de problemas durante a competição.
- Uma boa dinâmica de grupo é tão essencial quanto o conhecimento técnico dos competidores individuais.

Por quê?

3. Mercado de Trabalho

- Grandes empresas valorizam a participação em competições de programação.
- As últimas Fases Nacionais da Maratona, por exemplo, foram patrocinadas por empresas como Google, Microsoft e B2W.
- A Google organiza sua própria competição de programação, a Google Code Jam, voltada para identificar talentos para um potencial emprego na mesma.

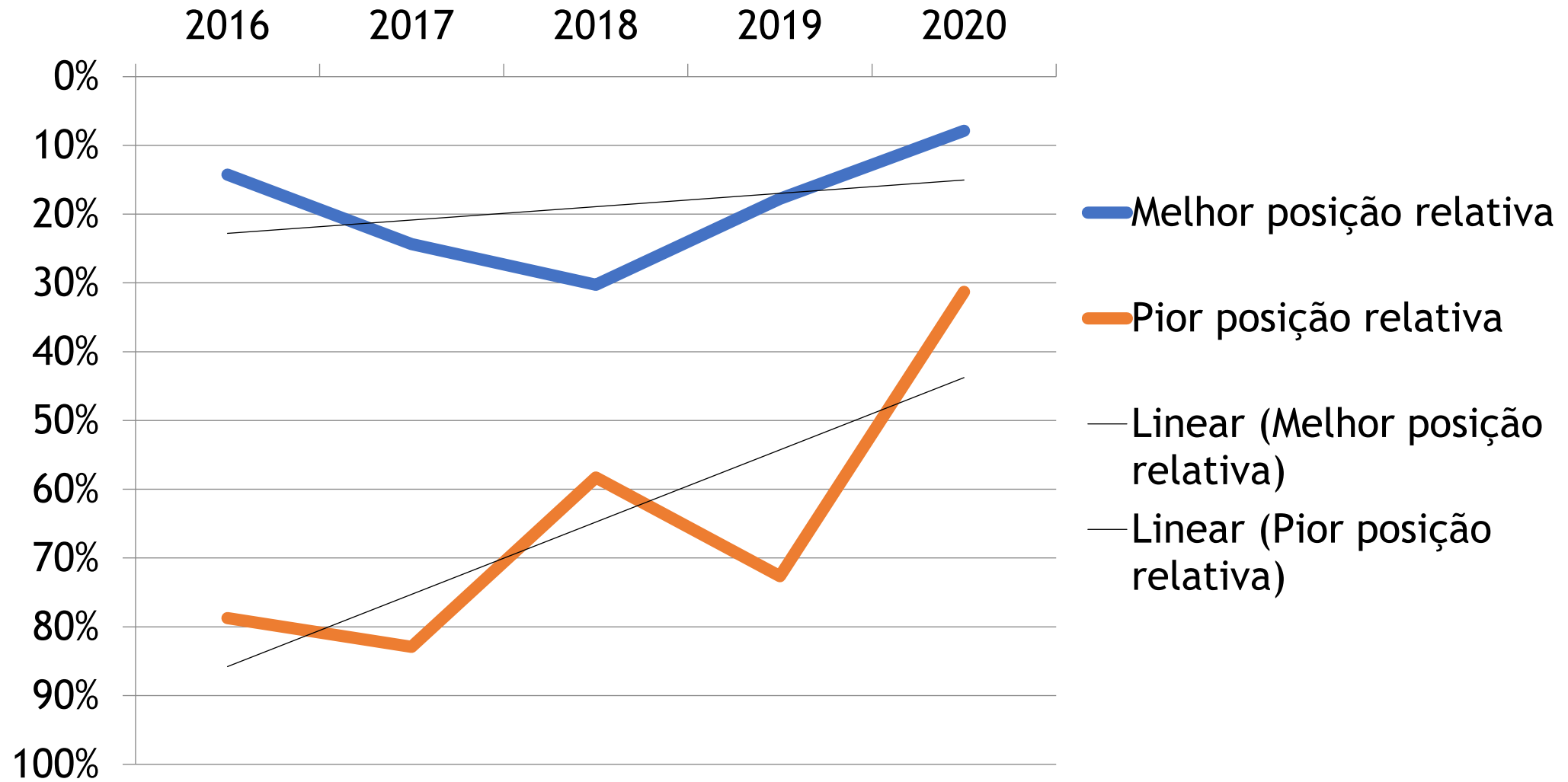
Por quê?

4. Viagens
 - Passeios
 - Comida
5. *Networking*
6. Brindes
7. Balões



“Observe que ser bem versado em programação competitiva não é objetivo final, apenas o meio. A verdadeira meta é produzir programadores/cientistas da computação versáteis, que estejam muito mais preparados para produzir *softwares* melhores ou para enfrentar complicados problemas de pesquisa de Ciência da Computação no futuro”

(Steven & Felix Halim, tradução nossa)



LPC I

- Aulas por videoconferência nas terças-feiras, das 19h às 23h.
 - Aulas teóricas sobre os principais temas cobrados na Maratona.
 - As aulas serão gravadas e disponibilizadas no YouTube ([Programação Competitiva UNESP](#))
 - Material complementar disponibilizado no Moodle
- *Contests* semanais
 - Toda semana ocorrerá uma competição, com a maioria dos exercícios relacionados ao tema da última aula.
 - Duração: de terça até domingo
 - Quantidade de problemas: de 5 a 7
 - Alternância entre *contests* individuais ou em grupo de até 3 pessoas
 - A formação das equipes pode variar ao longo do semestre, bastando nos avisar

LPC I

- Simulados

- Ao longo do semestre realizaremos competições com duração de 4 a 5 horas de duração, com problemas sobre os temas trabalhos até o momento.
- Serão realizadas aos sábados.
- Previsão de até três simulados ao longo do semestre.

Ferramentas e linguagem de programação

- Sistema operacional Linux
- VS Code
- Linguagem C++
- Compilador G++
- Sites:
 - [uDebug](#)
 - [Codeforces](#)
 - [URI Online Judge](#)
 - [Vjudge](#)
- Máquina virtual UNESP.ova (para quem não tem Linux instalado)

LPC I - Programação das atividades

1. Introdução e aquecimento
2. Introdução ao C++ STL - Sort, vector, queue e stack
3. Busca binária - C++ STL (lower_bound, upper_bound, map, set, priority_queue)
4. Força Bruta - Backtracking
5. Algoritmo Guloso - Divisão e Conquista
6. Programação Dinâmica I
7. Programação Dinâmica II

LPC I - Programação das atividades

8. Árvores
9. Disjoint-set (union-find)
10. Introdução à Teoria dos Grafos
11. Grafos: Problema do Caminho Mínimo
12. Grafos: Árvore Geradora Mínima (MST)
13. Strings: STL, KMP e palíndromos
14. Teoria dos Números
15. Análise combinatória

LPC II???

1. Busca ternária
2. Grafos: Pontes e Ordenação Topológica
3. Grafos: Fluxo Máximo
4. Grafos: Problema da Coloração e Emparelhamento Máximo
5. Strings: Suffix Trie, Tree & Array
6. Geometria computacional
7. Digit-DP

LPC II???

- 8. Programação Dinâmica com Bitmask e Bitset
- 9. PD: Convex Hull Trick
- 10. Teoria dos Jogos
- 11. Segment Tree
- 12. Árvore de Fenwick (BIT)
- 13. Sparse Table
- 14. HLD Decomposition
- 15. LCA

Bibliografia

- Felix Halim, Steven Halim. **Competitive Programming 3.**
- Steven Skiena. **Programming Challenges: The Programming Contest Training Manual.**
- Antti Laaksonen. **Guide to Competitive Programming: Learning and Improving Algorithms Through Contests**
- Steven Skiena. **The Algorithm Design Manual.**

Outros recursos

- [CP-Algorithms](#): uma enciclopédia de diversos algoritmos em C++, com tópicos bem explicados.
- [Geeks for Geeks](#): diversos artigos sobre programação e computação como um todo.
- [Neps Academy](#): contém cursos, problemas e plataformas para discussão. Diversos conteúdos possuem acesso gratuito, outros necessitam a realização de uma assinatura.

Outros recursos

- Canais do YouTube:
 - [GEMA ICMC](#): canal do Grupo de Estudos para a Maratona de Programação (GEMA) do ICMC - USP São Carlos.
 - [MaratonUSP](#): canal do grupo de estudos do IME-USP.