Olimpíadas Nacionais de Informática

Pedro Ribeiro (DCC/FCUP – Universidade do Porto)

OLIMPÍADAS NACIONAIS DE INFORMÁTICA

APDSI

U.PORTO FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

PRINCETON UNIVERSITY

CIÊNCIA VIVA
Agência Nacional para a Cultura Científica e Tecnológica

FUNDAÇÃO CALOUSTE GULBENKIAN

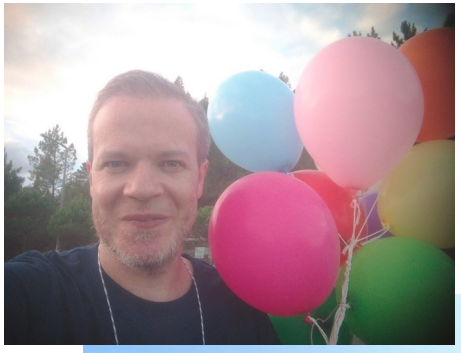
axians

Com o Alto Patrocínio de Sua Excelência
O Presidente da República

REPÚBLICA PORTUGUESA
EDUCAÇÃO, CIÊNCIA E INOVAÇÃO

anpri
Associação Nacional de Professores de Informática

Pedro Ribeiro - Quem Sou Eu?



Professor na **Universidade do Porto**

- Diretor do **Mestrado em Ciência de Computadores**
- Investigação em **Algoritmos** e em **Ciência de Redes** *(grafos)*

Pedro Ribeiro - Quem Sou Eu?

- Estou há muitos anos ligado a **concursos de programação**

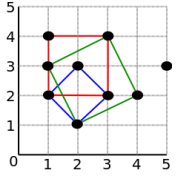
Como **participante** (comecei no 9º ano)
(2º lugar ONI'95, 1º lugar nas ONI 96, 97 e 98)
[representei Portugal nas IOI por 4 vezes]

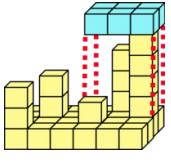
Como **mentor/treinador** e **líder** **(de delegação)**
(lidero a delegação portuguesa nas IOI desde 2025)
[lidero e treino equipas da minha universidade]

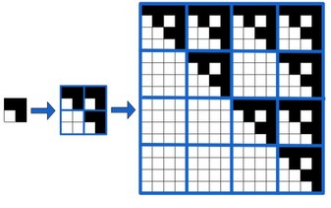
Como **organizador**
(final no "meu" DCC desde 2006)
[inúmeros outros concursos para várias faixas etárias]

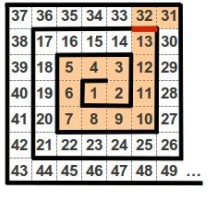
Bebras PT

Como **criador de problemas**
(centenas de problemas usados em eventos nacionais e inernacionais)









Porquê?


PORTUGAL


ioi
THE INTERNATIONAL OLYMPIAD IN INFORMATICS
LONG SERVICE AWARD
PRESENTED TO
Pedro Ribeiro
CELEBRATING 15 YEARS OF PARTICIPATION
Presented During
IOI 2024 Egypt
Benjamin Burton
President of IOI


THE 36TH INTERNATIONAL
OLYMPIAD IN INFORMATICS
ALEXANDRIA
EGYPT 2024
PORTUGAL

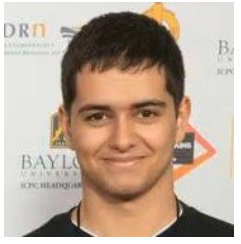
Missão das Olimpíadas de Informática

- **Captar** e **Desenvolver talento** na Informática

José Santos (IOI'98)
Principal Software Engineer
at **Microsoft**

João Oliveirinha (IOI'03)
Sr Engineering Manager
at **Cloudflare**

André Santos (IOI'06)
Software Enginner
at **Google**



Gonçalo Paredes
(IOI'14,15,16)
Sr. Engineer
at **TripAdvisor**

Pedro Paredes
(IOI'11,12)
Professor na
Univ. Princeton

Pedro Dias (IOI'19,20)
U. Cambridge

Rui Whang (IOI'20)
U. Oxford

Tiago Marques (IOI'21,22)
MIT

- Promover o gosto pelos **algoritmos** e pela **programação**
- Selecionar e preparar os concorrentes portugueses para as **IOI** (e **WEOI**, **OII** e **EGOI**)
- Uma **comunidade** para quem gosta destes temas

ONI (Olimpíadas Nacionais de Informática)

- ONI já vão na sua **38ª edição**

- Selecionam para as **IOI** **(Olimpíadas Internacionais de Informática)**

  - uma das grandes Olimpíadas Internacionais de Ciências
  - 4ª mais antiga, desde 89 (IMO desde 59, IPhO desde 67, IChO desde 68)

IOI (International Olympiad in Informatics)

- As Olimpíadas são uma **experiência marcante**

Fases das ONI 2026

OLIMPÍADAS NACIONAIS DE INFORMÁTICA

Inscrições — Até 15 de março *(online)*

C/C++, Java, Python

Qualificação — 18, 19, 20 de março *(local + online)*

30 participantes

C/C++, Java, Python

Final Nacional — 18 de abril *(presencial, DCC/FCUP)*

8 a 10 participantes

C++

Prova de Seleção — final de maio *(presencial, local a definir)*

6 participantes

C++

WEOI

Luxemburgo

26-28 junho *(presencial)*

4 participantes

C++

UZBEKISTAN

IOI 2026 TASHKENT

9-16 agosto *(presencial)*

todos da seleção

C++

Ibero-American Olympiad in Informatics

junho/julho *(remoto)*

Fases das ONI 2026

Com patrocínio extra de:

axians

OLIMPÍADAS NACIONAIS DE INFORMÁTICA

Inscrições — Até 15 de março *(online)*

C/C++, Java, Python

Qualificação — 18, 19, 20 de março *(local + online)*

30 participantes
+ 6 meninas

C/C++, Java, Python

Final Nacional — 18 de abril *(presencial, DCC/FCUP)*

2 a 4 meninas

C++, Python

EGOI 2026
Cesenatico, Italy
12-18 maio *(presencial)*

8 a 10 participantes
+ 2 a 4 meninas

C++

Prova de Seleção — final de maio *(presencial, local a definir)*

6 participantes
Potencialmente
+ 2 meninas

C++

WEOI
Luxemburgo
26-28 junho *(presencial)*

4 participantes

C++

UZBEKISTAN
IOI 2026 TASHKENT
9-16 agosto *(presencial)*

todos da seleção

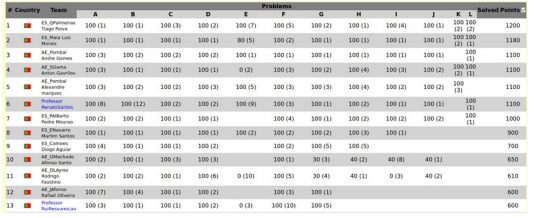
C++

Ibero-American Olympiad in Informatics
junho/julho *(remoto)*

(aumentar participação feminina!)

O sistema de submissão usado nas ONI

- Nas ONI usamos o **Mooshak** *(que é semelhante a outros plataformas usadas em programação competitiva)*



#	Country	Team	Problems												Solved	Points
			A	B	C	D	E	F	G	H	I	J	K	L		
1		ES_OPalmeiras Tiago Paiva	100 (1)	100 (1)	100 (3)	100 (2)	100 (7)	100 (5)	100 (2)	100 (1)	100 (4)	100 (1)	100 (2)	100		1200
2		ES_Maia Luis Morais	100 (1)	100 (1)	100 (1)	100 (1)	80 (5)	100 (2)	100 (1)	100 (1)	100 (1)	100 (1)	100 (2)	100 (1)		1180
3		AE_Pombal Andre Gomes	100 (3)	100 (2)	100 (2)	100 (2)	100 (1)	100 (1)	100 (1)	100 (1)	100 (1)	100 (1)	100 (1)			1100
4		AE_SGoma Anton Gavrilov	100 (3)	100 (1)	100 (1)	100 (1)	0 (2)	100 (3)	100 (2)	100 (4)	100 (3)	100 (2)	100 (2)	100 (1)		1100
5		AE_Pombal Alexandre marques	100 (3)	100 (2)	100 (2)	100 (3)	100 (5)	100 (3)	100 (3)	100 (4)	100 (2)	100 (2)	100 (3)			1100
6		Professor RenatoSantos	100 (8)	100 (12)	100 (2)	100 (2)	100 (9)	100 (3)	100 (1)	100 (2)	100 (1)	100 (1)	100 (1)			1100
7		ES_PAlberto Pedro Mourao	100 (2)	100 (2)	100 (1)	100 (1)		100 (4)	100 (1)	100 (2)	100 (2)	100 (2)	100 (1)			1000
8		ES_FAlmeida Martim Santos	100 (1)	100 (1)	100 (1)	100 (1)	100 (2)	100 (2)	100 (2)	100 (3)	100 (1)					900
9		ES_Carneiro Diogo Aguiar	100 (4)	100 (1)	100 (1)	100 (2)		100 (2)	100 (5)	100 (5)						700
10		AE_DMachado Afonso Sante	100 (2)	100 (1)	100 (3)	100 (3)		100 (1)	30 (3)	40 (2)	40 (8)	40 (1)				650
11		AE_DLaynes Rodrigo Faustino	100 (2)	100 (2)	100 (1)	100 (6)	0 (10)	100 (5)	30 (4)	40 (1)	0 (3)	40 (2)				610
12		AE_JAlbino Rafael-Oliveira	100 (7)	100 (4)	100 (1)	100 (2)		100 (3)	100 (1)							600
13		Professor RuiRenascimento	100 (3)	100 (1)	100 (1)	100 (2)	0 (3)	100 (10)	100 (5)							600

- Na sua essência:
  - **Submetemos** código-fonte
  - O programa é automaticamente **compilado** e **executado**
  - As respostas dadas pelo vosso programa para um **conjunto de testes** são comparadas com as respostas esperadas
  - A **pontuação** depende dos testes onde “passaram”

O sistema de submissão usado nas ONI

- Vamos agora fazer uma pequena **demonstração interativa** do seu uso, incluindo a obtenção dos seguintes resultados:
  - Accepted (teste aceite)
  - Compile Time Error (o programa não compila)
  - Presentation Error (existem problemas com *"whitespace"*)
  - Wrong Answer (resposta errada)
  - Runtime Error (programa "crasha" durante a execução)
  - Time Limit Exceeded (o problema excedeu o tempo limite)

Moshak

S887	738:00:54		AE_Arraiolos Diogo Esteves	B	C++	100 Accepted	final
S886	737:56:37		AE_Arraiolos Diogo Esteves	A	C++	100 Accepted	final
S885	737:53:29		AE_Arraiolos Diogo Esteves	A	C++	0 Presentation Error	final
S884	737:49:42		AE_GMachado Prikshit Sahni	B	C	84 Wrong Answer	final
S883	737:48:17		AE_GMachado Prikshit Sahni	B	C	52 Wrong Answer	final
S882	737:47:10		AE_GMachado Prikshit Sahni	B	C	0 Compile Time Error	final
S881	737:32:02		AE_GMachado Miguel Henriques	L	Python	100 Accepted	final
S880	737:30:43		AE_GMachado Miguel Wheeler	D	C++	0 Compile Time Error	final
S879	737:29:41		AE_GMachado Miguel Wheeler	D	C++	0 Compile Time Error	final
S878	737:29:14		AE_GMachado Miguel Henriques	L	Python	40 Time Limit Exceeded	final

Anatomia de um problema de Olimpíadas

- Os problemas são de **caracter algorítmico**
  - Não chega estar *"correto"*, é preciso ser **eficiente** ao nível de **tempo** e **memória**

- Os problemas podem ter **várias subtarefas**:
  - Restrições diferentes que admitem soluções diferentes (ex: menos eficientes, casos específicos, etc)

- Existem vários tipos de problemas
  - "Batch Problems" (clássicos) – Input / Output
  - Output-Only; Interação; ...

Exemplo de Problema

Problemas anteriores disponíveis no nosso website: https://oni.dcc.fc.up.pt/problemas

- Vejamos um **exemplo** de problema (Qualificação'2020)

  https://www.dcc.fc.up.pt/oni/problemas/2020/qualificacao/probA.html

Problema A - Corridas de Berlindes

Este é problema usa o **novo formato de problemas**.
Consulta a página de instruções para informações detalhadas sobre como funciona este novo formato.

Apesar de pouco falado, poucos desportos têm ganho mais espetadores do que as corridas de berlindes (em especial as Olimpíadas da Nutação Intensiva). Neste desporto há **N** berlindes que competem entre si para ver quem é o mais rápido. Como responsável por organizar as corridas das ONI, não podes perder esta oportunidade para mostrar aos novos fãs o melhor que o desporto tem para dar! Para tal, queres organizar as mais diversas e entusiasmantes corridas - os fãs não gostam que as corridas sejam iguais! Neste caso, a diversidade mede-se pelas características dos berlindes: o *i*-ésimo berlinde tem uma agilidade $\mathbf{A_i}$ e uma velocidade $\mathbf{V_i}$, ambos números inteiros.

Parte I

Para manter a competição interessante queres evitar berlindes iguais, ou seja, com a mesma agilidade e a mesma velocidade. Isto garante que as corridas são o mais variadas possível. Como responsável pela organização, queres saber, dado os teus **N** berlindes, quantos berlindes diferentes tens?

Exemplo

Se **N** = 5 e os berlindes tiverem as seguintes agilidades e velocidades, respetivamente: (1, 2), (3, 4), (1, 2), (1, 4), (3, 2), então há 4 berlindes diferentes porque há dois berlindes com agilidade 1 e velocidade 2.

- **Input:** **N** berlindes, cada um com uma agilidade A_i e uma velocidade V_i

Exemplo de Problema

Solução foi abordada no webinar – explicação detalhada disponível em:
https://oni.dcc.fc.up.pt/loop/solucoes/2020/qualificacao/prob_a.html

- **Parte I**

  Quantos berlindes **diferentes** existem?

  - Subtarefa 1: N ≤ 1 000
  - Subtarefa 2: N ≤ 100 000

  Solução bruta em $O(N^2)$: verificar todos os pares

  Solução eficiente:

  - Em $O(N)$: criar array bidimensional *(restrição chave: $0 \leq A_i, V_i \leq 10$)*
  - Em $O(N \log N)$: colocar num **`set`** (existe equiv. em C++/Java/Python)

Exemplo de Problema

Solução foi abordada no webinar – explicação detalhada disponível em:
https://oni.dcc.fc.up.pt/loop/solucoes/2020/qualificacao/prob_a.html

- **Parte II**

  Se todos combatem contra todos e um berlinde ganha a outro quando tem simultaneamente melhor agilidade e velocidade, **quantas vitórias tem o "campeão"**?

  - Subtarefa 1: N ≤ 1 000
  - Subtarefa 2: N ≤ 100 000

  Solução bruta em $O(N^2)$: verificar todos os pares

  Solução eficiente em $O(N \times B)$: criar array bidimensional

  - onde B é o número de possíveis berlindes (no máximo são... 121)

    (restrição chave novamente é $0 \leq A_i, V_i \leq 10$)

Exemplo de Problema

Solução foi abordada no webinar – explicação detalhada disponível em:
https://oni.dcc.fc.up.pt/loop/solucoes/2020/qualificacao/prob_c.html

- Vejamos um **outro exemplo** de problema (Qualificação'2020)

  https://www.dcc.fc.up.pt/oni/problemas/2020/qualificacao/probC.html

Problema C - Rua das Flores

Este é problema usa o **novo formato de problemas**.
Consulta a página de instruções para informações detalhadas sobre como funciona este novo formato.

A Rua das Flores é uma rua muito longa (tem **N** metros de comprimento!) e que há muito tempo está em terríveis condições. No entanto, e felizmente para os condutores que percorrem essa rua frequentemente, o presidente da freguesia prometeu fazer obras nessa rua para ela deixar de ser um problema no quotidiano dos seus transeuntes.

A rua pode ser vista como uma sequência **S** de **N inteiros positivos**, onde o *i*-ésimo, $\mathbf{S_i}$, corresponde ao custo de arranjar o *i*-ésimo metro da rua. Fazer obras em toda a rua pode exceder o orçamento da junta de freguesia, por isso estão a ser estudadas 3 opções distintas para a obra.

Parte I

Uma das formas de cumprir o plano eleitoral é fazer obras de recuperação num **segmento contíguo** de **S** de comprimento exatamente **K** metros. Recordem que um segmento contíguo é um intervalo de elementos seguidos, por exemplo para a sequência [1, 2, 2, 3] a subsequência [1, 2, 2] é uma subsequência contígua (que contém os primeiros três elementos) mas [1, 2, 3] já não (contém o primeiro, segundo e quarto elementos). Entre todos os segmentos contíguos com **K** metros de comprimento, pretende-se descobrir aquele que tem o custo total mínimo de recuperação.

- No webinar iremos apenas falar da parte I
  *(e do conceito de **sliding window**)*

Analisando a Eficiência

- Para **analisar a eficiência** usamos uma notação (**Big O**) que serve para **caracterizar a complexidade** de um **algoritmo**:
  - Plataforma Loop:
    - https://oni.dcc.fc.up.pt/loop/guias/inicial/efi/
    - https://oni.dcc.fc.up.pt/loop/guias/base/introalg/
  - Aulas Universitárias:
    - Slides de uma aula minha (e um vídeo meu)
    - Big O Cheat Sheet
  - Livros:
    - Principles of Algorithmic Problem Solving (chapter 5)
    - Competitive Programmer's Handbook (chapter 2)

Recursos de Treino

- Como me posso então **preparar/treinar**?

  Um **misto** entre:

  - **Aprender** novas coisas
    - Complexidade, algoritmos, estruturas de dados
    - Livros, videos, blogs
  - **Resolver problemas** fora de uma prova
    - Pré-categorizados para cimentar conhecimento
    - Sem saber a categoria *(problem solving)*
  - **Participar em provas**
    - Participar em provas com tempo a contar
    - No final ler editoriais/fazer *upsolving*

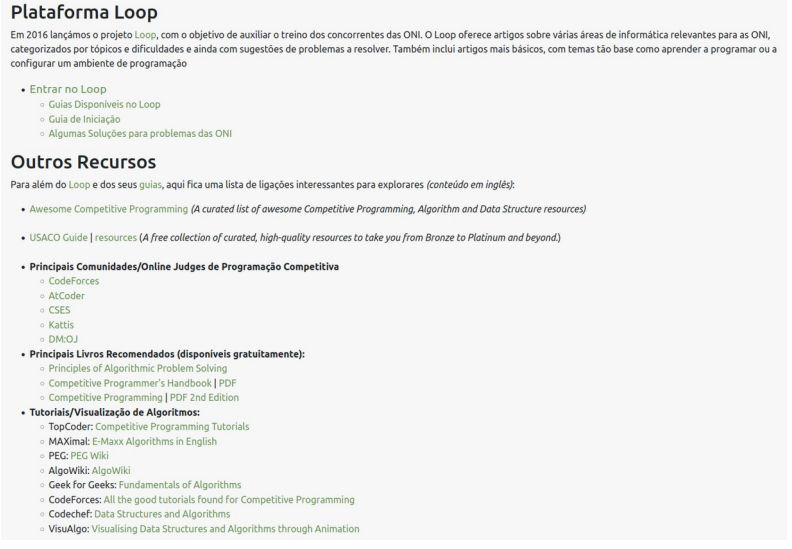
Não existe “varinha mágica” – trabalhar, trabalhar!

Recursos de Treino

- Página online já contém ligações para muitos **recursos de treino**

  (quase que existe *"informação a mais"* no universo da Programação Competitiva)

https://oni.dcc.fc.up.pt/plataforma_treino



Plataforma Loop

Em 2016 lançámos o projeto Loop, com o objetivo de auxiliar o treino dos concorrentes das ONI. O Loop oferece artigos sobre várias áreas de informática relevantes para as ONI, categorizados por tópicos e dificuldades e ainda com sugestões de problemas a resolver. Também inclui artigos mais básicos, com temas tão base como aprender a programar ou a configurar um ambiente de programação

- Entrar no Loop
  - Guias Disponíveis no Loop
  - Guia de Iniciação
  - Algumas Soluções para problemas das ONI

Outros Recursos

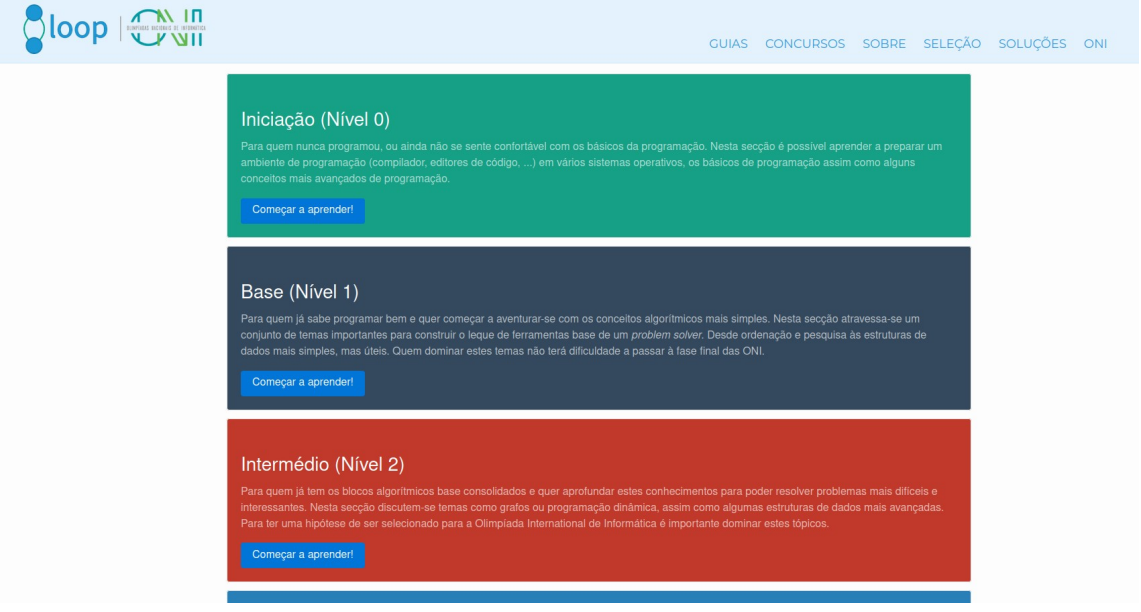
Para além do Loop e dos seus guias, aqui fica uma lista de ligações interessantes para explorares *(conteúdo em inglês)*:

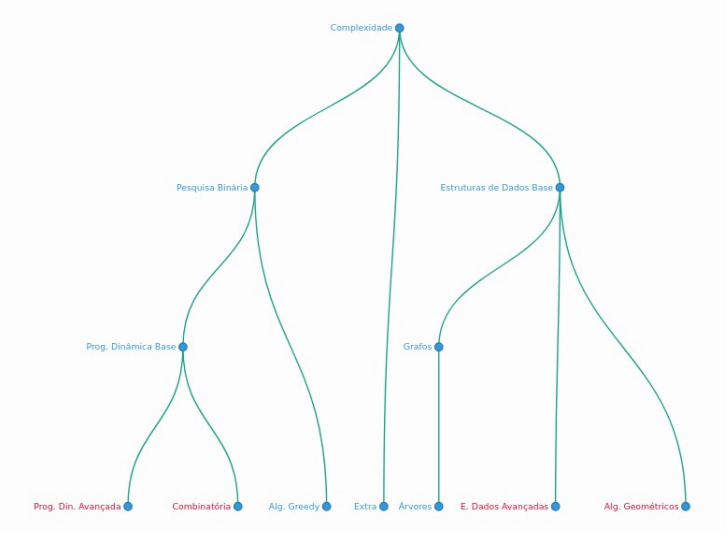
- Awesome Competitive Programming *(A curated list of awesome Competitive Programming, Algorithm and Data Structure resources)*
- USACO Guide | resources *(A free collection of curated, high-quality resources to take you from Bronze to Platinum and beyond.)*
- **Principais Comunidades/Online Judges de Programação Competitiva**
  - CodeForces
  - AtCoder
  - CSES
  - Kattis
  - DM:OJ
- **Principais Livros Recomendados (disponíveis gratuitamente):**
  - Principles of Algorithmic Problem Solving
  - Competitive Programmer's Handbook | PDF
  - Competitive Programming | PDF 2nd Edition
- **Tutoriais/Visualização de Algoritmos:**
  - TopCoder: Competitive Programming Tutorials
  - MAXimal: E-Maxx Algorithms in English
  - PEG: PEG Wiki
  - AlgoWiki: AlgoWiki
  - Geek for Geeks: Fundamentals of Algorithms
  - CodeForces: All the good tutorials found for Competitive Programming
  - Codechef: Data Structures and Algorithms
  - VisuAlgo: Visualising Data Structures and Algorithms through Animation

Recursos de Treino - Loop

- Conjunto de Guias e Recursos de Treino

https://oni.dcc.fc.up.pt/loop/





Recursos de Treino - Livros

- **Principles of Algorithmic Problem Solving**

https://jsannemo.se/latest.pdf

Contents

Recursos de Treino - Livros

- **Competitive Programmer's Handbook**

https://cses.fi/book/book.pdf

Contents

Recursos de Treino - Livros

- **Competitive Programming**

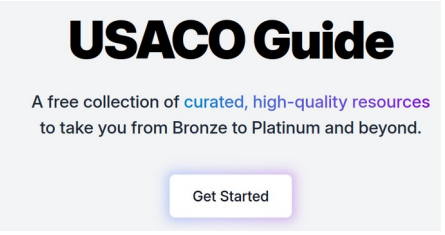
https://www.comp.nus.edu.sg/~stevenha/myteaching/competitive_programming/cp2.pdf

Contents

Recursos de Treino - Wikis/Tutoriais/Visualização

- **https://usaco.guide/**

  programa de treino integrado americano
  (altamente recomendado)



- **https://cp-algorithms.com/**

  repositório de implementações (C++)
  de algoritmos com explicação

- **Wikis/Tutorials:**
  - https://www.geeksforgeeks.org/dsa/dsa-tutorial-learn-data-structures-and-algorithms/
  - https://wiki.algo.is/
  - https://wcipeg.com/wiki/Special:AllPages
  - https://codeforces.com/blog/entry/57282

Recursos de Treino – Concursos/Problemsets

- **Codeforces:** https://codeforces.com/

  A maior comunidade de programação competitiva com concursos regulares (pode-se fazer concursos anteriores e ver editoriais)

- **AtCoder:** https://atcoder.jp/

- **CSES:** https://cses.fi/problemset/

- **DM:OJ**: https://dmoj.ca/problems/?search=IOI

Recursos de Treino – Quero saber mais?

https://oni.dcc.fc.up.pt/plataforma_treino

- https://github.com/lnishan/awesome-competitive-programming

  A curated list of awesome Competitive Programming, Algorithm and Data Structure resources.

- **IOI Syllabus**: https://algo.sk/ioi-syllabus/

The International Olympiad in Informatics Syllabus

1 Version and status information

IOI 2024 is using the current official Syllabus version (unchanged since March 2022). This is a draft of the Syllabus version intended for IOI 2025. At the moment no changes other than those already included in this document are expected. At the end of IOI 2024 this document should become the official Syllabus.

The Syllabus aims to achieve these goals by providing a classification of topics and concepts from mathematics and computer science. More precisely, this Syllabus classifies each topic into one of six categories. Ordered by topic suitability, these are:

- ✓ Included, unlimited
- ✓ Included, to be defined
- ✓ Included, not for task description
- ? Outside of focus
- ✗ Excluded, but open to discussion
- ✗ Explicitly excluded

Final da Apresentação

- Obrigado pela atenção! :)

- **Contactos:**
  - Website: https://oni.dcc.fc.up.pt/
  - Email: oni@dcc.fc.up.pt
  - Discord: https://discord.com/invite/c8B7WBEQeF