

# Maratona SBC de Programação: 2025

## — Participação da UVV (v. 1.0) —

Abrantes Araújo Silva Filho

2025-05-15

### Resumo

Este documento informa as regras para a participação de times de alunos da UVV na Maratona SBC de Programação de 2025, o programa de treinamento e estudos e as políticas de inclusão e exclusão de times.

## Sumário

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Participação no Treinamento</b>	<b>3</b>
2.1	Quem pode participar? . . . . .	3
2.2	Alunos e times . . . . .	4
2.3	Vagas . . . . .	4
2.4	Dinâmica do treinamento . . . . .	5
2.5	Políticas de exclusão . . . . .	8
<b>3</b>	<b>Participação na Maratona</b>	<b>8</b>
<b>4</b>	<b>O que eu ganho com isso?</b>	<b>9</b>
	<b>Referências</b>	<b>9</b>

# 1 Introdução

A Maratona de Programação é um evento da Sociedade Brasileira de Computação (SBC) que existe desde o ano de 1996, tendo sido originada das competições regionais classificatórias para as etapas mundiais da competição de programação, o *International Collegiate Programming Contest* (ICPC), e é parte da super-regional latino americana do evento ([SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 2025](#)).

A Universidade Vila Velha (UVV) participou da Maratona SBC de Programação de 2024 com 3 equipes<sup>1</sup> e a equipe “Tiger Devs” obteve a 6ª colocação entre todas as equipes do Espírito Santo (Figura 1):

Figura 1: Resultado regional: ES

#	User/Site	Name	A	B	C	D	E	F	G	H	I	J	K	L	Total
1	teambresvi005/BR	[UFES] Machine Burning	1/8				1/38	1/34		9/-	1/-		1/-	1/62	4 (134)
2	teambresvi004/BR	[UFES] try catch us	1/9	1/-			1/33	1/18	4/-					3/116	4 (216)
3	teambresvi013/BR	[UFES] Óscura	1/8	3/-			1/82	1/53					1/-	1/93	4 (236)
4	teambresvi008/BR	[FAESA] Vue Só	1/8				1/56	1/36					1/-	8/-	3 (100)
5	teambresvi003/BR	[UFES] big 0(no)	2/25				1/68	3/76	1/-				2/-	1/-	3 (229)
6	teambresvi007/BR	[UVV] Tiger Devs	3/32				1/75	1/215							3 (362)
7	teambresvi006/BR	[IFES] Error 451: Unavailable for Legal Reasons	1/8				2/156	1/195		6/-	2/-				3 (379)
8	teambresvi012/BR	[FAESA] undefined	1/9				2/53	9/-		8/-					2 (82)
9	teambresvi001/BR	[UVV] PythonKillers	1/19				1/63							2/-	2 (82)
10	teambresvi002/BR	[UVV] How do I close Vim	1/45				1/65	2/-							2 (110)
11	teambresvi009/BR	[IFES] Dark Mode	1/16				5/91								2 (187)

Essa 6ª colocação foi uma vitória pessoal dos alunos da UVV e um ótimo resultado final, pois as equipes consistiam basicamente de alunos dos períodos iniciais da computação (a maioria dos alunos ainda nem tinha feito as disciplinas de Estruturas de Dados I e II), os alunos tiveram apenas três ou quatro semanas para realizar um treinamento muito básico sobre resolução de problemas computacionais complexos, e a infra-estrutura para os treinamento estava aquém do necessário para um bom aproveitamento dos estudos. Mais ainda: as equipes nas três posições iniciais, formadas por alunos mais experientes da Universidade Federal do Espírito Santo (UFES), acertaram apenas 1 problema a mais do que a equipe “Tiger Devs” (veja algumas fotos da Maratona de 2024 em <https://disciplinas.uvv.br/2024/09/01/maratona/>).

Na maratona de 2025, que será realizada no dia 13 de setembro, nossa meta é figurar entre as três posições superiores do ranking, alcançando no mínimo a medalha de bronze.

<sup>1</sup>Na realidade 5 equipes se inscreveram mas 2 equipes não compareceram à competição, causando grandes transtornos para a UVV.

## 2 Participação no Treinamento

A UVV, através de professores voluntários, oferecerá um **treinamento intensivo** de 14 semanas, entre os dias 31/05/2025 e 06/09/2025, com uma semana final de preparação antes da data da maratona (13/09/2025). Esse treinamento englobará diversos tópicos como algoritmos e estruturas de dados, resolução de problemas, programação competitiva e outros. Esta seção detalha as regras de participação no treinamento (inclusão, exclusão e políticas).

### 2.1 Quem pode participar?

A participação dos alunos da área de computação da UVV é aberta, ou seja, todos os alunos interessados podem participar, desde que sejam cumpridos os seguintes requisitos:

- **REQUISITOS OBRIGATÓRIOS.** Para a participação nos treinamentos da maratona é necessário que o aluno tenha os seguintes requisitos:
  - **Comprometimento e dedicação.** O treinamento para a maratona será intensivo, durante 14–15 semanas, com aulas presenciais aos sábados durante o período letivo e aulas remotas aos sábados durante as férias<sup>2</sup>. A cada semana cada aluno deverá cumprir uma série grande de exercícios e tarefas de programação, além de ter que estudar tópicos avançados em estruturas de dados e algoritmos. Por favor: se você não está disposto a fazer esse sacrifício intenso durante essas 15 semanas, não se inscreva para o treinamento, não ocupe a vaga de algum outro aluno mais comprometido e dedicado. **ATENÇÃO:** o treinamento terá **políticas de exclusão** para desligar os alunos que não se comprometerem e/ou se dedicarem o suficiente.
  - O aluno já deve ter feito a **disciplina de programação básica** de seu curso (por exemplo: Laboratório de Programação em C ou Python) e deve já ter cursado, ou estar cursando, a disciplina inicial de **estruturas de dados** de seu curso (por exemplo: Estrutura de Dados I).
  - O aluno deve saber programar em pelo menos uma das seguintes **linguagens**: C, C++, Python, Java, Kotlin. Idealmente o aluno deveria saber programar em duas dessas linguagens, por exemplo: C++ e Python.
  - O aluno deve ter iniciado seus estudos há no máximo 5 anos **OU** completar no máximo 23 anos de idade em 2025 (essa é uma regra da SBC).
  - O aluno deve saber **ler em inglês**, pois os exercícios de treinamento e simulação são escritos nessa língua<sup>3</sup>.

---

<sup>2</sup>Sim, o curso não será interrompido durante as férias.

<sup>3</sup>A maratona oficial terá os problemas escritos em português.

- O aluno deve ter um entendimento de **análise simples de algoritmos**, no nível de uma disciplina de Estrutura de Dados (no mínimo deve compreender as complexidades dos algoritmos através da notação Big-O).
- **Requisitos desejáveis:**
  - O aluno deve ter completado, ou estar cursando, alguma disciplina de **matemática discreta** ou equivalente.
  - Idealmente o aluno deve ter terminado um curso mais avançado de algoritmos e estruturas de dados (Estrutura de Dados II).
  - Familiaridade com Linux (a maratona é realizada em uma máquina Linux especialmente criada para a competição, sem acesso à Internet).
  - Familiaridade com ferramentas de debug tais como GDB e Valgrind.

## 2.2 Alunos e times

Da mesma forma que a participação na maratona, a participação nos treinamentos é feita por **times de 3 alunos**, ou seja, você não pode se inscrever sozinho. Se você quiser participar você deve **montar um time** de três alunos que cumpram os requisitos listados na Seção 2.1, principalmente quanto ao comprometimento e dedicação aos estudos e treinamentos.

Se um **integrante desistir no meio do treinamento**, especialmente após a terceira ou quarta semana, **todo o time poderá ser excluído do programa**. É importante que você monte seu time apenas com alunos que você conheça e tenha certeza de que irão se comprometer durante as 14–15 semanas de treinamento intensivo.

Sabe aquele aluno que, nos trabalhos em grupo nunca faz nada e é sempre carregado pelos outros? Ou aquele aluno que passa colando em todas as provas e ainda se vangloria de ter enganado o professor? Ou aquele aluno que você sabe que não se esforça nos exercícios de programação e copia tudo do ChatGPT ou pega o código de outros alunos? Esse tipo de aluno é mortal para uma equipe de programação competitiva... Vá à forra e elimine esse tipo de aluno de sua equipe, mesmo que seja seu amigo ou irmão. Ele será um peso morto e fará sua equipe afundar.

## 2.3 Vagas

Para a **participação no treinamento**, até 5 times de 3 alunos poderão se inscrever. Esses times se comprometerão a cumprir todas as atividades do curso intensivo preparatório para a maratona (14–15 semanas).

Para a **participação na maratona**, dependendo do desempenho individual dos alunos e do desempenho global de cada time, os professores e a UVV poderão recomendar: a) a participação de todos os times; b) a exclusão de times específicos, por baixo comprometimento e desempenho; c) a realização de uma pré-maratona para a seleção dos melhores times; d) outras situações.

A inscrição dos times será feita através de um formulário online a ser divulgado no momento apropriado. As equipes também assinarão um formulário de comprometimento e dedicação, concordando com os critérios de exclusão.

## 2.4 Dinâmica do treinamento

Os treinamentos para a maratona serão realizados **aos sábados**, de forma presencial no período letivo e de forma remota no período de férias, sempre de 08:00h até 12:00h, no Laboratório de Hardware.

O cronograma previsto de atividades está ilustrado na Figura 2 (na página 6). Por favor, note que esse cronograma pode sofrer ajustes e alterações no conteúdo conforme o andamento do treinamento e o aproveitamento pelos alunos. De qualquer forma, mesmo que ocorram ajustes no conteúdo, os treinamentos serão mantidos aos sábados, exceto em situações excepcionais.

Nos treinamentos aos sábados os professores farão uma revisão dos tópicos de algoritmos e estruturas de dados mais importantes para a maratona e, também, abordarão tópicos mais avançados que não costumam ser vistos por alunos em cursos de graduação. Além disso, aos sábados, cada equipe também realizará tarefas e exercícios de programação relativos aos conteúdos estudados. Por favor note que a **participação nos treinamentos é obrigatória**. Alunos que deixarem de comparecer aos treinamentos podem ser excluídos do treinamento.

Além dos treinamentos aos sábados os alunos devem, de forma individual e também como um time, resolver diversos exercícios de programação, na semana subsequente à aula de treinamento. Por exemplo: no dia 31/05 será ministrada a aula de “Introdução e Técnicas de Programação” e, entre os dias 01/06 té 06/06, os alunos devem completar todos os exercícios e tarefas de programação referentes ao conteúdo da aula.

Essa dinâmica, “aula → exercícios”, será seguida durante todo o período do treinamento. A **entrega dos exercícios é obrigatória** e será realizada através de sistemas online que realizam a verificação das respostas conforme as regras da maratona. Os sistemas utilizados serão os seguintes:

- Kattis (<https://open.kattis.com>)
- UVa Online Judge (<https://onlinejudge.org>)
- BOCA UFES (<https://boca.pet.inf.ufes.br/boca>)

O Kattis e o UVa serão utilizados para a resolução dos exercícios e tarefas de programação referentes ao conteúdo ministrado nas aulas, ou seja: após cada aula o professor indicará exatamente quais exercícios os alunos devem realizar. A realização de todos os exercícios é **obrigatória e individual**: todos os alunos de um mesmo time devem fazer os exercícios, sem copiar uns dos outros. Alunos que deixarem de entregar essas tarefas podem ser excluídos do treinamento.

O BOCA UFES será utilizado para simular o ambiente computacional real da maratona. Periodicamente o professor indicará um conjunto de 10 a 13 problemas

Figura 2: Cronograma previsto de atividades

Preparação para a Maratona de Programação 2025																								
Abril							Maio							Junho										
s	do	se	te	qu	qu	se	sá	s	do	se	te	qu	qu	se	sá	s	do	se	te	qu	qu	se	sá	
13				1	2	3	4	5	17					1	2	3	22	1	2	3	4	5	6	7
14	6	7	8	9	10	11	12	18	4	5	6	7	8	9	10	23	8	9	10	11	12	13	14	
15	13	14	15	16	17	18	19	19	11	12	13	14	15	16	17	24	15	16	17	18	19	20	21	
16	20	21	22	23	24	25	26	20	18	19	20	21	22	23	24	25	22	23	24	25	26	27	28	
17	27	28	29	30				21	25	26	27	28	29	30	31	26	29	30						
Julho							Agosto							Setembro										
s	do	se	te	qu	qu	se	sá	s	do	se	te	qu	qu	se	sá	s	do	se	te	qu	qu	se	sá	
26				1	2	3	4	5	30					1	2	35								
27	6	7	8	9	10	11	12	31	3	4	5	6	7	8	9	36	7	8	9	10	11	12	13	
28	13	14	15	16	17	18	19	32	10	11	12	13	14	15	16	37	14	15	16	17	18	19	20	
29	20	21	22	23	24	25	26	33	17	18	19	20	21	22	23	38	21	22	23	24	25	26	27	
30	27	28	29	30	31			34	24	25	26	27	28	29	30	39	28	29	30					
								35	31															
Cronograma PREVISTO:																								
Dia 31/05: Introdução e Técnicas de Programação																	(LAB. HARDWARE)							
Semana 22: Entrega dos exercícios e do BOCA																								
Dia 07/06: Estruturas de Dados e Tipos Abstratos de Dados																	(LAB. HARDWARE)							
Semana 23: Entrega dos exercícios e do BOCA																								
Dia 14/06: Algoritmos de Ordenação e Busca																	(LAB. HARDWARE)							
Semana 24: Entrega dos exercícios e do BOCA																								
Semana 25: Entrega dos exercícios e do BOCA																								
Dia 28/06: Paradigmas de Resolução de Problemas																	(REMOTO)							
Semana 26: Entrega dos exercícios e do BOCA																								
Dia 05/07: Programação Dinâmica																	(REMOTO)							
Semana 27: Entrega dos exercícios e do BOCA																								
Dia 12/07: Backtracking																	(REMOTO)							
Semana 28: Entrega dos exercícios e do BOCA																								
Dia 19/07: Matemática I																	(REMOTO)							
Semana 29: Entrega dos exercícios e do BOCA																								
Dia 26/07: Matemática II																	(REMOTO)							
Semana 30: Entrega dos exercícios e do BOCA																								
Dia 02/08: Strings																	(LAB. HARDWARE)							
Semana 31: Entrega dos exercícios e do BOCA																								
Dia 09/08: Algoritmos para Grafos																	(LAB. HARDWARE)							
Semana 32: Entrega dos exercícios e do BOCA																								
Dia 16/08: Design de Algoritmos																	(LAB. HARDWARE)							
Semana 33: Entrega dos exercícios e do BOCA																								
Dia 23/08: Algoritmos para Árvores																	(LAB. HARDWARE)							
Semana 34: Entrega dos exercícios e do BOCA																								
Dia 30/08: Grids																	(LAB. HARDWARE)							
Semana 35: Entrega dos exercícios e do BOCA																								
Dia 06/09: Geometria																	(LAB. HARDWARE)							
Semana 36: Entrega dos exercícios e preparação final																								
Excluído : Tópicos Avançados e Raros																								
Dia 13/09: MARATONA DE PROGRAMAÇÃO																	(IFES Serra)							

do BOCA que o **time completo** deve resolver: três alunos utilizando um único computador com o ambiente padronizado (Maratona Linux). Times que não realizarem as simulações podem ser excluídos do treinamento.

Por fim, cada aluno/time deve ser responsável por **estudar os livros e materiais** indicados pelos professores, referentes aos conteúdos da semana. Serão utilizados os seguintes livros:

- Livros específicos sobre programação competitiva:
  - *Competitive Programming*, volume 1 ([HALIM; HALIM; EFFENDY, 2020a](#))
  - *Competitive Programming*, volume 2 ([HALIM; HALIM; EFFENDY, 2020b](#))
  - *Guide to Competitive Programming* ([LAAKSONEN, 2024](#))
  - *Programming Challenges* ([SIKENA; REVILLA, 2003](#))
- Livros de consulta e referência:
  - *Algorithms* ([SEdgeWICK; WAYNE, 2011](#))
  - *Introduction to Algorithms* ([CORMEN et al., 2022](#))
  - *The Algorithm Design Manual* ([SKIENA, 2020](#))
  - *Algorithms in C* ([SEdgeWICK, 1990](#))
  - *Algorithms in C (Parts 1–4)* ([SEdgeWICK, 1998a](#))
  - *Algorithms in C (Part 5)* ([SEdgeWICK, 2002](#))
  - *Algorithms in C++ (Parts 1–4)* ([SEdgeWICK, 1998b](#))
  - *Algorithms in C++ (Part 5)* ([SEdgeWICK, 2001](#))
  - *Algorithm Design* ([KLEINBERG; TARDOS, 2006](#))
  - *Algoritmos* ([DASGUPTA; PAPADIMITRIOU; VAZIRANI, 2009](#))
  - *Algorithms Illuminated. Part 1: the basics* ([ROUGHGARDEN, 2017](#))
  - *Algorithms Illuminated. Part 2: graph algorithms and data structures* ([ROUGHGARDEN, 2018](#))
  - *Algorithms Illuminated. Part 3: greedy algorithms and dynamic programming* ([ROUGHGARDEN, 2019](#))
  - *Algorithms Illuminated. Part 4: algorithms for NP-Hard problems* ([ROUGHGARDEN, 2020](#))
  - *Computers and Intractability* ([GAREY; JOHNSON, 1979](#))
  - *The Art and Science of C* ([ROBERTS, 1995](#))
  - *Programming Abstractions in C* ([ROBERTS, 1998](#))
  - *Programming Abstractions in C++* ([ROBERTS, 2013](#))
  - *Thinking Recursively* ([ROBERTS, 1986](#))
  - *The C Programming Language* ([KERNIGHAN; RITCHIE, 1988](#))
  - *Hacker's Delight* ([WARREN JR., 2012](#))

Os livros cujas licenças permitem distribuição (livros publicados em *Creative Commons* ou outra licença equivalente) serão entregues aos alunos via arquivo PDF. Os livros que não permitem distribuição devem ser consultados diretamente na biblioteca da UVV.

## 2.5 Políticas de exclusão

O treinamento será realizado em 14–15 semanas e será **intenso**, com uma carga de trabalho muito maior do que os alunos estão acostumados. Se você ou alguém do seu time não estiver comprometido e dedicado em participar desse esforço intenso, por favor, **não se inscreva** e deixe a vaga para outro aluno que está mais motivado ou tem mais tempo de participação neste ano. As maratonas são anuais, se você não puder participar neste ano, organize-se para participar no próximo ano.

Este treinamento adotará as seguintes políticas de exclusão de times:

- O aluno que faltar 5 (cinco) treinamentos aos sábados, consecutivos ou não, será excluído do treinamento, causando a exclusão automática de todo o seu time<sup>4</sup>.
- O aluno que, individualmente, deixar de realizar 30% dos exercícios obrigatórios no Kattis e/ou UVa, consecutivos ou não, será excluído do treinamento, causando a exclusão automática de todo o seu time.
- O time que deixar de realizar 30% das simulações obrigatórias no BOCA também será excluído do treinamento.

### Por que regras draconianas de exclusão?

Na maratona de 2025 tivemos 15 alunos (5 equipes) inscritas, que tiveram os custos de inscrição bancados pela UVV. Infelizmente 7 alunos não participavam ativamente e faltavam aos treinamentos. Pior ainda: sem avisar a ninguém esses 7 alunos simplesmente **não compareceram** no dia da maratona, causando a desclassificação automática de 3 times.

Para evitar ou diminuir a chance desse tipo de problema ocorrer novamente, essas políticas de exclusão foram estabelecidas e serão seguidas. É melhor que menos alunos e times participem, mas de forma realmente comprometida, do que muitos alunos e times que não estão realmente ligando para a maratona.

## 3 Participação na Maratona

Os alunos e times que conseguirem cumprir todo o período de treinamento estarão aptos para concorrer às vagas para a **participação na maratona**, dependendo do desempenho individual dos alunos e do desempenho global de cada time.

A seleção dos alunos e times será feita pelos professores da UVV que, usando critérios objetivos e subjetivos, poderão recomendar:

- A participação de todos os times; ou

<sup>4</sup>Essa regra não é negociável. O segredo aqui está em escolher pessoas REALMENTE comprometidas e dispostas a participar.



- A exclusão de alunos e times específicos; ou
- A realização de uma pré-maratona para a seleção dos melhores times; ou
- Alguma outra forma de decisão.

Os alunos e equipes selecionados terão os custos de inscrição bancados pela UVV. Também estamos providenciando transporte de ida e volta para o local da maratona.

## 4 O que eu ganho com isso?

A participação na maratona de programação é, por si mesma, um esporte empolgante, mas o maior benefício da participação da maratona não é ganhar uma medalha ou reconhecimento: o maior benefício é desenvolver suas habilidades de raciocínio algorítmico, sua capacidade de resolver problemas, sua intuição e competência diante de tarefas complexas e difíceis.

Após passar por todas as semanas de treinamento, realizar os exercícios e tarefas de programação você:

- Desenvolverá sua capacidade de raciocínio algorítmico
- Desenvolverá suas habilidades de resolução de problemas
- Desenvolverá sua intuição e competência na resolução de tarefas complexas
- Estará preparado para entrevistas técnicas nas “*big techs*”
- Terá compreensão profunda sobre estruturas de dados e algoritmos avançados
- Se tornará um cientista avançado da computação

A jornada será árdua mas, se completada, você nunca mais será o mesmo.

## Referências

CORMEN, T. H. et al. *Introductions to Algorithms*. 4. ed. [S.l.]: The MIT Press, 2022. ISBN 978-0-262-04630-5. Citado na página 7.

DASGUPTA, S.; PAPADIMITRIOU, C.; VAZIRANI, U. *Algoritmos*. 1. ed. São Paulo: McGraw Hill, 2009. ISBN 978-85-7726-032-4. Citado na página 7.

GAREY, M. R.; JOHNSON, D. S. *Computers and Intractability: a guide to the theory of NP-Completeness*. 1. ed. [S.l.]: Freeman, 1979. ISBN 0-7167-1045-5. Citado na página 7.

HALIM, S.; HALIM, F.; EFFENDY, S. *Competitive Programming (vol. 1)*. 4. ed. Lulu, 2020. v. 1. ISBN 979-1716745521. Disponível em: <https://cpbook.net/>. Citado na página 7.

HALIM, S.; HALIM, F.; EFFENDY, S. *Competitive Programming (vol. 2)*. 4. ed. Lulu, 2020. v. 2. ISBN 979-1716745514. Disponível em: <https://cpbook.net/>. Citado na página 7.

KERNIGHAN, B. W.; RITCHIE, D. M. *The C Programming Language*. 2ª. ed. New Jersey: Prentice Hall, 1988. ISBN 9780131103627. Citado na página 7.

KLEINBERG, J.; TARDOS, É. *Algorithm Design*. 1. ed. [S.l.]: Addison-Wesley, 2006. ISBN 0-321-29535-8. Citado na página 7.

LAAKSONEN, A. *Guide to Competitive Programming: learning and improving algorithms through contests*. 3. ed. [S.l.]: Springer, 2024. ISBN 978-3-031-61793-5. Citado na página 7.

ROBERTS, E. S. *Thinking Recursively*. 1. ed. John Wiley and Sons, 1986. ISBN 978-0471816522. Disponível em: <https://cs.stanford.edu/people/eroberts/books/ThinkingRecursively/index.html>. Citado na página 7.

ROBERTS, E. S. *The Art and Science of C: A library based introduction to computer science*. Massachusetts: Addison Wesley, 1995. ISBN 9780201543223. Citado na página 7.

ROBERTS, E. S. *Programming Abstractions in C*. 1. ed. [S.l.]: Addison-Wesley, 1998. ISBN 0-201-54541-1. Citado na página 7.

ROBERTS, E. S. *Programming Abstractions in C++*. 1. ed. Prentice Hall, 2013. ISBN 978-0133454840. Disponível em: [https://web.stanford.edu/dept/cs\\_edu/resources/textbook/](https://web.stanford.edu/dept/cs_edu/resources/textbook/). Citado na página 7.

ROUGHGARDEN, T. *Algorithms Illuminated. Part 1: the basics*. 1. ed. Soundlikeyourself Publishing, 2017. ISBN 978-0-9992829-0-8. Disponível em: <https://www.algorithmsilluminated.org/>. Citado na página 7.

ROUGHGARDEN, T. *Algorithms Illuminated. Part 2: graph algorithms and data structures*. 1. ed. Soundlikeyourself Publishing, 2018. ISBN 978-0-9992829-2-2. Disponível em: <https://www.algorithmsilluminated.org/>. Citado na página 7.

ROUGHGARDEN, T. *Algorithms Illuminated. Part 3: greedy algorithms and dynamic programming*. 1. ed. Soundlikeyourself Publishing, 2019. ISBN 978-0-9992829-4-6. Disponível em: <https://www.algorithmsilluminated.org/>. Citado na página 7.

ROUGHGARDEN, T. *Algorithms Illuminated. Part 4: algorithms for NP-Hard problems*. 1. ed. Soundlikeyourself Publishing, 2020. ISBN 978-0-9992829-6-0. Disponível em: <https://www.algorithmsilluminated.org/>. Citado na página 7.

SEDGEWICK, R. *Algorithms in C*. 1. ed. [S.l.]: Addison-Wesley, 1990. ISBN 0-201-51425-7. Citado na página 7.

SEDGEWICK, R. *Algorithms in C (Parts 1–4)*. 3. ed. [S.l.]: Addison-Wesley, 1998. v. 1. ISBN 978-0-201-31452-6. Citado na página 7.

- SEDGEWICK, R. *Algorithms in C++ (Parts 1–4)*. 3. ed. [S.l.]: Addison-Wesley, 1998. ISBN 978-0201350883. Citado na página 7.
- SEDGEWICK, R. *Algorithms in C++ (Part 5)*. 3. ed. [S.l.]: Addison-Wesley, 2001. ISBN 978-0201361186. Citado na página 7.
- SEDGEWICK, R. *Algorithms in C (Part 5)*. 3. ed. [S.l.]: Addison-Wesley, 2002. v. 2. ISBN 0-201-31663-3. Citado na página 7.
- SEDGEWICK, R.; WAYNE, K. *Algorithms*. Fourth. Addison Wesley, 2011. ISBN 9780321573513. Disponível em: <https://algs4.cs.princeton.edu/home/>. Citado na página 7.
- SIKENA, S. S.; REVILLA, M. A. *Programming Challenges: the programming contest training manual*. 1. ed. [S.l.]: Springer, 2003. ISBN 0-387-00163-8. Citado na página 7.
- SKIENA, S. S. *The Algorithm Design Manual*. 3. ed. Springer, 2020. ISBN 978-3-030-54255-9. Disponível em: <https://www.algorist.com/>. Citado na página 7.
- SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. *O que é a Maratona SBC de Programação*. 2025. Internet. [Acesso em: 2025-05-15]. Disponível em: <https://maratona.sbc.org.br/sobre/index.html>. Citado na página 2.
- WARREN JR., H. S. *Hacker's Delight*. 2. ed. [S.l.]: Addison-Wesley, 2012. ISBN 978-0133085013. Citado na página 7.