

Técnicas para Programação Competitiva

Introdução à Programação Competitiva

Prof. Andrei Braga



Conteúdo

- Apresentação da turma
- Programação competitiva
- Competições
- Plataforma CSES
- Tarefa
- Referências

Conteúdo

- **Apresentação da turma**
- Programação competitiva
- Competições
- Plataforma CSES
- Tarefa
- Referências

Apresentação da turma

- Qual é o seu nome?
- Você faz a graduação em qual turno (vespertino ou noturno)?
- No momento, além de cursar disciplinas, você está trabalhando/estagiando/engajado(a) em algum projeto em uma área relacionada à computação?
- Com quais linguagens de programação você tem mais experiência?
- Você já ouviu falar/participou da Maratona de Programação?
- Você já resolveu problemas em plataformas online (ex.: Beecrowd)? Se sim, quais?
- Onde você passa mais tempo: YouTube, Instagram ou TikTok? Tem ou já teve um canal/perfil com muitos seguidores? 😎

Conteúdo

- Apresentação da turma
- **Programação competitiva**
- Competições
- Plataforma CSES
- Tarefa
- Referências

Programação competitiva

Ideia: Resolver problemas conhecidos de Ciência da Computação o mais rápido possível! (HALIM, S.; HALIM, F.; EFFENDY, S., 2018)

- **Problemas conhecidos** são problemas para os quais já se conhece uma solução – ao contrário de problemas de pesquisa, para os quais ainda está se buscando uma solução

Programação competitiva

Ideia: Resolver problemas conhecidos de Ciência da Computação o mais rápido possível! (HALIM, S.; HALIM, F.; EFFENDY, S., 2018)

- Para **resolver** estes problemas, precisamos aprofundar e aplicar os nossos conhecimentos de computação com o intuito de construir um programa que, para um conjunto de testes secreto,
 - determine a resposta correta para cada teste e
 - execute dentro do tempo limite permitido

Programação competitiva

Ideia: Resolver problemas conhecidos de Ciência da Computação o mais rápido possível! (HALIM, S.; HALIM, F.; EFFENDY, S., 2018)

- O elemento **competitivo** está em dois objetivos:
 - Resolver o maior número possível de problemas
 - Fazer isto o mais rápido possível

Problema

Descrição adaptada do problema “10911 - Forming Quiz Teams” do UVa Online Judge

- Sejam (x, y) as coordenadas inteiras da casa de um estudante em um plano 2D. Existem $2N$ estudantes e queremos agrupá-los em N grupos. Para $i = 1, 2, \dots, N$, seja $d(i)$ a distância entre as casas dos 2 estudantes do grupo i . Devemos formar os N grupos de modo que a soma $d(1) + d(2) + \dots + d(i)$ seja a menor possível. Restrições: $1 \leq N \leq 8$ e $0 \leq x, y \leq 1000$.
- **Entrada:** N e as coordenadas das $2N$ casas dos estudantes.
- **Saída:** O menor valor possível da soma $d(1) + d(2) + \dots + d(i)$, impresso com 2 casas decimais.

Problema

Descrição adaptada do problema “10911 - Forming Quiz Teams” do UVa Online Judge

- Sejam (x, y) as coordenadas inteiras da casa de um estudante em um plano 2D. Existem $2N$ estudantes e queremos agrupá-los em N grupos. Para $i = 1, 2, \dots, N$, seja $d(i)$ a distância entre as casas dos 2 estudantes do grupo i . Devemos formar os N grupos de modo que a soma $d(1) + d(2) + \dots + d(i)$ seja a menor possível. Restrições: $1 \leq N \leq 8$ e $0 \leq x, y \leq 1000$.
- Exemplo de entrada:** $N = 3$ e as coordenadas das 6 casas são $(1, 2)$, $(1, 8)$, $(3, 3)$, $(3, 7)$, $(4, 4)$ e $(4, 6)$.
- Exemplo de saída:** 6.47
- Dica: use este [site](#) para facilitar os cálculos



Abordagens para (tentar) resolver o problema

- Abordagem 1
 - Formar um grupo com os 2 estudantes cujas casas têm a menor distância entre si
 - Repetir o passo anterior até formar todos os grupos
 - O algoritmo composto por estes passos é um **algoritmo guloso**
 - Resultado: **Wrong Answer** (Resposta Errada)
- Abordagem 2
 - Considerar todas as maneiras possíveis de formar N grupos
 - Isto pode ser implementado através de um algoritmo de **backtracking** simples (isto, é, sem otimizações)
 - Resultado: **Time Limit Exceeded** (Tempo Limite Excedido)

Abordagens para (tentar) resolver o problema

- Abordagem 3
 - Perceber que este é um problema de emparelhamento em um grafo
 - Considerando que não sabemos resolver este tipo de problema ainda...
 - Resultado: **Pular para o próximo problema** 😄
- Abordagem 4
 - Perceber que este é um problema de emparelhamento em um grafo e que é um problema difícil
 - No entanto, sabendo que o tamanho da entrada é pequeno (veja *Restrições* na descrição do problema), usar um algoritmo de **programação dinâmica**
 - Resultado: **Accepted** (Solução Aceita)

Abordagens para (tentar) resolver o problema

- Abordagem 5
 - Depois de adquirir mais experiência, programar a solução da Abordagem 4 mais rapidamente
- Abordagem 6
 - Depois de adquirir ainda mais experiência, programar a solução da Abordagem 4 ainda mais rapidamente e já conhecer antecipadamente soluções para variantes do problema

Programação competitiva - Objetivos

- Ser bom em programação competitiva **não** é o objetivo principal
- O **objetivo principal** é contribuir para a formação de profissionais capazes de
 - produzir software de qualidade,
 - resolver problemas difíceis de computação (problemas conhecidos e futuramente problemas de pesquisa),
 - trabalhar em equipe e
 - atuar sob pressão

Programação competitiva - Dicas para aprender

- Melhorar as nossas habilidades em programação competitiva requer dedicação – precisamos praticar, praticar e praticar
- No entanto, existem maneiras de praticar melhores do que outras
- Mais importante do que a **quantidade** de problemas que vamos resolver é a **qualidade** destes problemas
 - Não é muito útil permanecer resolvendo problemas fáceis, sempre evitando problemas difíceis
- Muitos problemas podem ser resolvidos com algoritmos simples – é importante aprender maneiras de lidar com problemas difíceis **usando ferramentas simples**
- Melhorar a habilidade de **ler em inglês** é muito importante – vários materiais e plataformas interessantes estão disponíveis apenas em inglês

Conteúdo

- Apresentação da turma
- Programação competitiva
- **Competições**
- Plataforma CSES
- Tarefa
- Referências

Padrão comum de competições de programação

- Formato de competição e recursos
 - Os competidores participam em equipes (geralmente, 3 pessoas)
 - Cada equipe tem acesso a apenas 1 computador
 - As equipes não têm acesso à internet e podem consultar apenas material impresso (livros, apostilas, compilado de material próprio e outros)
 - A duração da competição é comumente de 5 horas



Imagem: [Equipe da UPF](#), disponível em [notícia](#) publicada em www.upf.br

Padrão comum de competições de programação

- Resolução de problemas
 - Cada equipe recebe o mesmo conjunto de problemas (sem indicação de ordem de dificuldade)
 - Quando considerar que programou corretamente uma solução para um problema, a equipe deve submeter o seu programa para avaliação
 - O programa submetido será executado e avaliado por um juiz automatizado, que testará o programa com o conjunto secreto de testes do problema
 - O resultado de uma submissão será um dos seguintes:
 - AC – Accepted (Solução Aceita)
 - WA – Wrong Answer (Resposta Errada)
 - TLE – Time Limit Exceeded (Tempo Limite Excedido)
 - CE – Compilation Error (Erro de Compilação)
 - RE – Runtime Error (Erro de Execução)

Não há informações sobre quais linhas do programa geraram erro ou em quais testes o programa falhou

Padrão comum de competições de programação

- Pontuação

- A cada problema resolvido, a equipe recebe um ponto – ou um balão (da cor correspondente ao problema)
- Ganha a competição a equipe que receber mais pontos



Imagem: [Maratona de Programação](http://maratona.sbc.org.br), via maratona.sbc.org.br

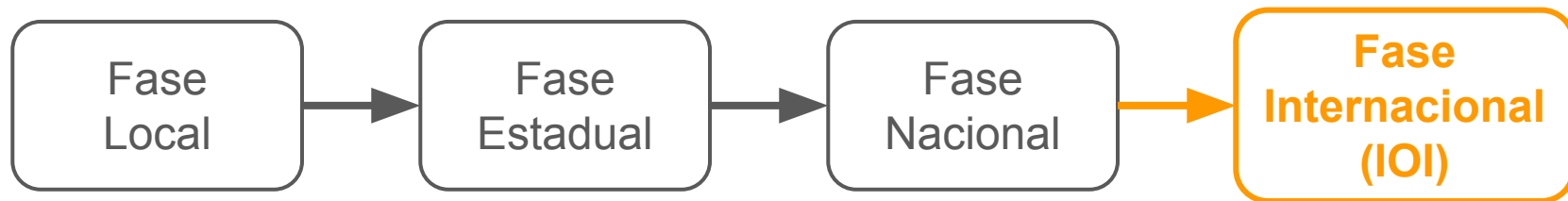
Padrão comum de competições de programação

- Pontuação

- Caso mais de uma equipe obtenha o mesmo número de pontos, o desempate é realizado de acordo com o menor tempo total de resolução dos problemas
- Submissões erradas geram penalidades de tempo
- O tempo associado a cada problema resolvido é calculado da seguinte forma:
 $\text{nº de submissões erradas} * 20 \text{ min} + \text{tempo (em min) para resolver o problema}$
- Geralmente, na primeira hora de prova, as equipes se dedicam a identificar os problemas mais fáceis e resolvê-los rapidamente sem submissões erradas

Olimpíada Brasileira de Informática (OBI)

- Podem participar estudantes do ensino fundamental, médio, técnico ou do primeiro ano do ensino superior
- Organizada pelo Instituto de Computação da UNICAMP
- Duas modalidades: Iniciação e Programação
- Três fases:



Olimpíada Internacional de Informática (IOI)

- Participam 4 estudantes brasileiros selecionados através da OBI
- O Brasil coleciona 42 medalhas na IOI: 3 de ouro, 11 de prata e 28 de bronze

Maratona SBC de Programação

- Podem participar estudantes que estejam fazendo graduação ou estejam no início da pós-graduação (de acordo com algumas regras)
- As equipes são formadas por 3 membros titulares e 1 reserva (opcional) e 1 treinador
- Duas fases:



Maratona SBC de Programação - Campeões Nacionais

- 10 vezes:



- 4 vezes:



- 3 vezes:



- 2 vezes:



- 1 vez:



Fonte: ref. 2 (veja o último slide)

Maratona SBC de Programação - Participação da UFFS

- Desde 2013, equipes da UFFS participaram 7 vezes da final nacional, obtendo as seguintes colocações:
 - 2013: 27ª colocação (1ª colocação na Fase Regional)
 - 2014: 20ª e 49ª colocações (1ª e 2ª colocações na Fase Regional)
 - 2015: 33ª colocação (3ª colocação na Fase Regional)
 - 2016: 42ª colocação (2ª colocação na Fase Regional)
 - 2017: 57ª colocação (2ª colocação na Fase Regional)
 - 2018: 36ª colocação (2ª colocação na Fase Regional)
 - 2019: 40ª colocação (1ª colocação na Fase Regional)

Maratona SBC de Programação - Edição 2022/2023

- Placar final da Final Nacional:

🔍 Available scores:
Brasil

#	User/Site	Name	A	B	C	D	E	F	G	H	I	J	K	L	M	Total
1	teambrbr005/BR	[UFRJ] Lebenslangerschicksalsschatz	1/117	1/156	1/48	1/21	1/29	1/269		2/206	1/54		1/295	1/83	1/133	11 (1431)
2	teambrbr002/BR	[UFMG] Humuhumunukunukuapua'a	2/127	2/237	1/99	1/11	2/49		1/254	1/176	1/36	1/-	1/191	1/148	2/106	11 (1514)
3	teambrbr001/BR	[UNICAMP] Você beijaria Matheus Leal Viana?	1/148	1/122	1/198	1/6	2/90	1/83		1/237	1/101		2/270	1/-	1/59	10 (1346)
4	teambrbr003/BR	[UNICAMP] Inimigos do Hungaro	2/80	1/260	1/46	1/17	2/54	1/-		1/104	1/62		3/141	9/-	1/29	9 (873)
5	teambrbr008/BR	[IME] 12k Club	1/44		1/86	1/12	1/19	2/-	1/201	1/166	1/32	1/-			1/61	8 (621)
6	teambrbr011/BR	[UnB] FLAMENGO	1/65		2/140	1/14	1/37			1/201	1/79		1/-	2/274	1/107	8 (957)
7	teambrbr006/BR	[USP] MWNWMWNNWMWNN	1/129		3/126	1/27	2/112	8/-		1/-	1/79		3/272		1/39	7 (884)
8	teambrbr019/BR	[UFCG] A ascensão e queda do buldogue francês	2/169			1/13	1/78			3/252	2/91		1/144		1/104	7 (931)

Maratona SBC de Programação - Edição 2022/2023

- Equipe campeã:



A participação de **meninas** ainda é muito **pequena**!

Maratona SBC de Programação - Mercado de Trabalho

- Estudantes que têm um bom desempenho na competição obtêm uma boa visibilidade em relação a grandes empresas de áreas relacionadas a computação
- As habilidades citadas anteriormente são muito atraentes em candidatos a vagas de emprego
- Muitas empresas de desenvolvimento de software patrocinam a Maratona SBC de Programação e outras competições de programação
- Várias empresas de tecnologia contam com ex-maratonistas

International Collegiate Programming Contests (ICPC)

- Participam as equipes mais bem colocadas na Fase Nacional da Maratona SBC de Programação – a 1ª colocada e outras equipes segundo algumas regras
- Conta com a participação de mais de 3 mil universidades localizadas em 103 países de 6 continentes

ICPC - Participação brasileira

- Equipes de 23 universidades brasileiras já participaram do ICPC



Fonte: ref. 2 (veja o último slide)

ICPC - Campeões latino-americanos

- Placar: 14  11  1 
- Equipes brasileiras:

- 3 vezes campeões:



- 1 vez campeão:



Fonte: ref. 2 (veja o último slide)

Conteúdo

- Apresentação da turma
- Programação competitiva
- Competições
- **Plataforma CSES**
- Tarefa
- Referências

Plataforma CSES

- A plataforma CSES é uma das plataformas que vamos utilizar nesta disciplina
- Esta plataforma contém uma coleção de problemas que podemos usar para praticar programação competitiva:
<https://cses.fi/problemset/>
- Os problemas estão organizados em ordem de dificuldade
- Técnicas para resolver estes problemas são discutidas no livro da primeira referência básica da disciplina

Plataforma CSES - Primeiro problema

Problem 1068 - Weird Algorithm (descrito em inglês)

- Consider an algorithm that takes as input a positive integer n . If n is even, the algorithm divides it by two, and if n is odd, the algorithm multiplies it by three and adds one. The algorithm repeats this, until n is one. For example, the sequence for $n = 3$ is as follows:

$$3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$$

Your task is to simulate the execution of the algorithm for a given value of n .
Constraints: $1 \leq n \leq 10^6$.

- Input:** The only input line contains an integer n .
- Output:** Print a line that contains all values of n during the algorithm.

Plataforma CSES - Primeiro problema

Problem 1068 - Weird Algorithm (descrito em inglês)

- Consider an algorithm that takes as input a positive integer n . If n is even, the algorithm divides it by two, and if n is odd, the algorithm multiplies it by three and adds one. The algorithm repeats this, until n is one. For example, the sequence for $n = 3$ is as follows:

$3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$

Your task is to simulate the execution of the algorithm for a given value of n .
Constraints: $1 \leq n \leq 10^6$.

- Input example:**
3
- Output example:**
3 10 5 16 8 4 2 1

Plataforma CSES - Primeiro problema

- Este é um problema de simulação simples, que pode ser resolvido em C++ da seguinte maneira:

```
#include <iostream>
using namespace std;
int main() {
    int n;
    cin >> n;
    while (true) {
        cout << n << " ";
        if (n == 1) break;
        if (n%2 == 0) n /= 2;
        else n = n*3+1;
    }
    cout << "\n";
}
```

Tarefa: Vamos executar este código no nosso computador para o exemplo de entrada dado anteriormente

Observe que a saída do programa é exatamente a saída esperada! ✌

Plataforma CSES - Primeiro problema

Tarefa: Agora, vamos submeter este código na plataforma CSES

O resultado será o mostrado ao lado

Test results ▲

test	verdict	time	
#1	ACCEPTED	0.00 s	»»
#2	ACCEPTED	0.00 s	»»
#3	ACCEPTED	0.00 s	»»
#4	ACCEPTED	0.00 s	»»
#5	ACCEPTED	0.00 s	»»
#6	TIME LIMIT EXCEEDED	--	»»
#7	TIME LIMIT EXCEEDED	--	»»
#8	WRONG ANSWER	0.00 s	»»
#9	OUTPUT LIMIT EXCEEDED	0.00 s	»»
#10	WRONG ANSWER	0.00 s	»»
#11	ACCEPTED	0.00 s	»»
#12	ACCEPTED	0.00 s	»»
#13	ACCEPTED	0.00 s	»»
#14	ACCEPTED	0.00 s	»»

Plataforma CSES - Primeiro problema

O primeiro teste que resulta em TIME LIMIT EXCEEDED corresponde a $n = 138367$ (veja os detalhes do teste)

O que acontece quando executamos este código no nosso computador para esta entrada?

O problema é que n atinge um valor muito grande durante a execução, maior que o valor máximo de uma variável do tipo `int`

Test results ▲

test	verdict	time	
#1	ACCEPTED	0.00 s	»»
#2	ACCEPTED	0.00 s	»»
#3	ACCEPTED	0.00 s	»»
#4	ACCEPTED	0.00 s	»»
#5	ACCEPTED	0.00 s	»»
#6	TIME LIMIT EXCEEDED	--	»»
#7	TIME LIMIT EXCEEDED	--	»»
#8	WRONG ANSWER	0.00 s	»»
#9	OUTPUT LIMIT EXCEEDED	0.00 s	»»
#10	WRONG ANSWER	0.00 s	»»
#11	ACCEPTED	0.00 s	»»
#12	ACCEPTED	0.00 s	»»
#13	ACCEPTED	0.00 s	»»
#14	ACCEPTED	0.00 s	»»

Plataforma CSES - Primeiro problema

Podemos consertar o problema alterando o tipo da variável n para **long long**

Ao submeter o código alterado, vamos obter o resultado ao lado

Test results ▲

test	verdict	time	
#1	ACCEPTED	0.00 s	»»
#2	ACCEPTED	0.00 s	»»
#3	ACCEPTED	0.00 s	»»
#4	ACCEPTED	0.00 s	»»
#5	ACCEPTED	0.00 s	»»
#6	ACCEPTED	0.00 s	»»
#7	ACCEPTED	0.00 s	»»
#8	ACCEPTED	0.00 s	»»
#9	ACCEPTED	0.00 s	»»
#10	ACCEPTED	0.00 s	»»
#11	ACCEPTED	0.00 s	»»
#12	ACCEPTED	0.00 s	»»
#13	ACCEPTED	0.00 s	»»
#14	ACCEPTED	0.00 s	»»

Plataforma CSES - Primeiro problema

- Esta situação nos dá um exemplo de que, mesmo em códigos bem simples, podemos ter bugs
- A prática com a programação competitiva aprimora a nossa habilidade de escrever programas corretos em geral

Conteúdo

- Apresentação da turma
- Programação competitiva
- Competições
- Plataforma CSES
- **Tarefa**
- Referências

Tarefa

1. Ler (curta uma leitura agradável!) o livreto disponibilizado [aqui](#), que contém uma introdução aos assuntos envolvidos na Maratona SBC de Programação

Referências

- Esta apresentação é baseada nos seguintes materiais:
 1. Capítulo 1 do livro
HALIM, S.; HALIM, F.; EFFENDY, S. Competitive Programming 4: The Lower Bound of Programming Contests in the 2020s, book 1, chs. 1-4. Lulu, 2018.
 2. Apresentação de slides
FERREIRA, C. Final Brasileira 2022. 2023.
 3. Apresentação de slides
Maratona SBC de Programação – Apresentação. 2023.
 4. Capítulo 1 do livro
LAAKSONEN, A. Guide to Competitive Programming: Learning and Improving Algorithms Through Contests, 2. ed. Springer, 2020.