

Introdução à Programação Competitiva

Prof. Daniel Saad Nogueira Nunes

12 de novembro de 2018



Sumário

- 1 Introdução
- 2 Programação Competitiva
- 3 Divulgação



Sumário

1 Introdução



Introdução

- Programação Competitiva: é um *mind sport* que visa colocar os participantes para resolver problemas através de técnicas de programação, algoritmos e estruturas de dados.
- É uma prática bem vista por diversas empresas ao redor do mundo: Google, Facebook, Amazon, . . .
- Com a prática de Programação Competitiva, o participante adquire:
 - ▶ Proficiência em paradigmas de projeto de algoritmos e estrutura de dados.
 - ▶ Capacidade de analisar a dificuldade de um problema.
 - ▶ Trabalho em equipe.
 - ▶ Conhecimento acerca de linguagens de programação.



Introdução

- Algumas competências exercitadas:
 - ▶ Estruturas de dados.
 - ▶ Projeto por divisão e conquista.
 - ▶ Programação dinâmica.
 - ▶ Algoritmos gulosos.
 - ▶ Problemas NP-completos e NP-difíceis.
 - ▶ Busca exaustiva.
 - ▶ Teoria dos Grafos.
 - ▶ Teoria dos Jogos.
 - ▶ Geometria Computacional.
 - ▶ Teoria dos Números.
 - ▶ Autômatos.
 - ▶ Análise de Algoritmos.
 - ▶ ...



Programação Competitiva

- Você sabe como Google contrata os seus engenheiros de software?

► [Google Interview](#)



Maratona de Programação

- A competição mais famosa no Brasil de programação competitiva é a **Maratona de Programação**.
- Equipes do Brasil todo participam buscando uma classificação para o ICPC (International Collegiate Programming Contest).
- Atualmente: mais de 800 times de 232 escolas diferentes participando em 50 sedes.
- Dividida em duas fases: regional e nacional.



Maratona de Programação



Maratona IFB

- De modo a capacitar os nossos alunos a participar da Maratona de Programação.
- Maratona de Programação todo semestre!



Maratona IFB

- I Maratona de Programação IFB: 15 equipes, cerca de 45 participantes.
- II Maratona de Programação do IFB: 43 equipes, cerca de 120 participantes. Participação remota e presencial.



Maratona IFB



Sumário

2 Programação Competitiva



Programação Competitiva

- Para treinar para competições e aperfeiçoar o conhecimento nas diversas competências, podemos usar os **juízes online**.
- São plataformas computacionais que oferecem diversos problemas e que aceitam submissão de códigos.
- O juiz então compila o código e testa ele contra uma porção de casos de testes.
- Se seu código passa em todos os casos de teste, você ganha um **Accept**.



Programação Competitiva

- Alguns juízes online:
 - ▶ `urionlinejudge.com.br`
 - ▶ `codeforces.com`
 - ▶ `csacademy.com`



Programação Competitiva

- Todas essas plataformas oferecem um espaço para discussão sobre problemas.
- Desta forma, se você “empaca” em algum problema, pode pesquisar e discutir com os demais colegas.
- Ambiente totalmente favorável ao aprendizado.



Programação Competitiva

- Vamos pegar alguns problemas para ilustrar alguns dos problemas mais comuns na adaptação dos iniciantes.



Dicas para Programação Competitiva

► URI 1542



Dicas para Programação Competitiva

```
#include <stdio.h>
int main(void){
    int q,d,p;
    while(scanf("%d",&q) && q>0){
        scanf("%d %d",&d,&p);
        int n = (d*p*q/(p-q));
        if(n>1){
            printf("%d paginas\n",n);
        }
        else{
            printf("%d pagina\n",n);
        }
    }

    return 0;
}
```



Dicas para Programação Competitiva

- Repare que, não foi impresso nada que o problema não tenha pedido.
- Evite colocar coisas do tipo: “Digite o número P”.
- A entrada é separada da leitura. Você não precisa imprimir as respostas só depois que ler toda a entrada.
- Fique atento para a condição de parada.
- Fique atento para os limites do problema!



Dicas para Programação Competitiva

► URI 1471



Dicas para Programação Competitiva

- O método de entrada deste problema foi um pouco diferente.
- A leitura deverá ser feita até o EOF.



Dicas para Programação Competitiva

```
#include <stdio.h>

int identificadores_retornaram[10001];
int marcado[10001];

int main(void){
    int n,r;
    while(scanf("%d %d",&n,&r)!=EOF){
        int i;
        for(i=1;i<=n;i++){
            marcado[i] = 0;
        }
        for(i=0;i<r;i++){
            int id;
            scanf("%d",&id);
            marcado[id] = 1;
        }
        if(n==r){
            printf("*\n");
        }
        else{
            for(i=1;i<=n;i++){
                if(!marcado[i]){
                    printf("%d ",i);
                }
            }
            printf("\n");
        }
    }
    return 0;
}
```



Dicas para Programação Competitiva

- Dependendo do tamanho da entrada do problema, suas estruturas de dados devem ser alocadas em uma região de memória chamada *heap* e não na *stack*, onde residem as variáveis locais.
- Caso contrário, você poderá ter um *Stack Overflow*.
- Solução mais fácil: colocar as principais EDs como global.
- OBS: não conte isso para o seu prof. de engenharia de software.
- O objetivo aqui é fazer o problema passar!
- Claro que em projetos, as boas práticas de software devem ser seguidas.



Dicas para Programação Competitiva

- Durante a sua submissão, além do **Accept**, você pode receber outros vereditos.
- **Time Limit Exceeded (TLE)**: o seu código não conseguiu rodar dentro do tempo limite do problema. Provavelmente seu algoritmo está ineficiente.
- **Wrong Answer (WA)**: seu problema falhou em algum caso de teste.
- **Presentation Error (PE)**: o seu programa está correto, mas ele está imprimindo espaços ou linhas a mais (ou a menos).
- **Compilation Error (CE)**: erro de compilação. Seu programa não conseguiu ser compilado. Verifique se escolheu o compilador correto na plataforma.



Dicas para a Programação Competitiva

- Em um *contest*, ganha a equipe que fizer mais problemas.
- Em caso de empate, utiliza-se o critério de tempo acumulado.
- Por exemplo: se uma equipe acerta um problema aos 100 minutos de competição, é adicionado 100 ao tempo acumulado.
- Caso uma equipe receba um veredito que não seja **AC** e venha acertar o problema depois, uma penalidade de 20 minutos é acrescida ao tempo acumulado para cada tentativa incorreta.



Dicas para a Programação Competitiva

- Os *contests* são realizados em trio.
- Apenas um computador por equipe está disponível.
- As equipes devem bolar uma estratégia para otimizar o tempo na frente do computador.
- Material impresso pode ser levado, mas é proibido consultar a Internet.



Sumário

3 Divulgação



Maratona IFB

- III Maratona de Programação do IFB.
- Data: 24/11/2018 (sábado).
- Aberto para todas as equipes do DF e outros estados.
- Participação presencial/online.
- Nível: **Iniciante**.



Maratona DF

- <https://facebook.com/maratonaIF>.
- <https://facebook.com/maratonaDF>.
- <https://danielsaad.com/maratona>



Slack

► Slack



Mini-contest

► Prévia III Maratona de Programação do IFB



Perguntas

Perguntas?

Contato

daniel.nunes@ifb.edu.br

