

Lista de Exercícios 1

Maratona de Programação

1 Informações gerais

1. Embora os problemas sejam simples de se resolver, quem ta fazendo pela primeira vez, não é fácil. Fácil não significa Simples!
2. O grupo serve pra a gente debater as soluções, não existe solução certa ou errada, se resolveu o problema e recebeu Accepted, sua solução tá certa.
3. Tenta googlar as coisas em ingles, infelizmente programação competitiva não é um assunto muito abordado no Brasil, tem informação em alguns sites e tal, mas em ingles é mais fácil e rápido de achar a resposta pra seu problema.
4. Se tu conseguiu passar dos casos de teste, mas continua recebendo TLE ou WA, provavelmente seu algoritmo não ta robusto o suficiente e ta falhando nos problemas de fronteiras. Por exemplo: receba um inteiro $i, 0 < i < 10^{10}$. Se você recebe esse i como um int, no caso de fronteira, você vai ter problemas de overflow. Então, cuidado com isso.
5. Aproveitando a deixa da dica acima, int, long int e double são os tipos mais usados nos problemas (tanto em C++ como em Java).

2 Informações pra C++

1. Site para documentação: <http://www.cplusplus.com/reference/>
2. Seus novos melhores amigos serão: vector e string, esqueça os arrays simples. (Tá, não esqueça. Em alguns casos, ainda é melhor os arrays, mas prefira vector e string)
3. Não é porque você está usando uma linguagem que tem acesso a ponteiros e manipulação de memória que você deve usar isso (Lembre que a maioria dos problemas que você esta resolvendo da pra resolver com Python e Java. Você NÃO está programando para embarcados). A partir do C++11 foi introduzido uma funcionalidade chamada smart pointers

que faz o gerenciamento de memória melhor do que você, isso é só curiosidade, a parte que importa pra a gente é: Todos os containers (e string) tem uma coisa chamada Iterator, quando precisar de um ponteiro, use o Iterator. Vá na referência se informar sobre o que é isso.

4. Tem um novo for, introduzido a partir do C++11 (se eu não me engano) que se chama for-each, dá uma pesquisada aí. Usa ele com a palavra reservada "auto".
5. Muitas das coisas que você quer fazer com um for, while e tal... Dá pra fazer com uma linha de código passando uma função lambda. Muitos algoritmos já estão implementados na linguagem, por exemplo: `copy_if`, `transform`, `next_permutation`, `find...` tem muitos. Todos esses algoritmos serão muito úteis, procure resolver os problemas "from-scratch" e depois usando eles. Todos esses algoritmos são funções que recebem a posição inicial de um container, a posição final e uma função lambda que vai ser responsável pela lógica que esse algoritmo fará por debaixo dos panos.

6. Depois de aprender essas coisas mais "básicas" procure entender sobre stringstream, é extremamente útil para algumas coisas com manipulação de strings. Mas so se preocupe com isso no futuro. Sério. Em NENHUM dos problemas dessa lista será necessário usar esse objeto. So to comentando agora, por questão de curiosidade.
7. C++ é uma linguagem orientada a objetos, C++ NÃO É C! Tenha calma se você estiver com dificuldades pra executar alguns dos algoritmos mencionados acima, pra entender o que é uma função lambda e tal... O criador da linguagem em uma entrevista disse que, de 0-10, ele tem um conhecimento 8 na linguagem, não espere ter um conhecimento maior do que o criador da linguagem.

3 Informações pra Java

1. Java é muito poderoso, porém lento em alguns aspectos, I/O é uma delas. Tem uma forma de otimizar isso criando uma classe "FastReader", o código dela ta em um template que eu coloquei no grupo, depois coloco no meu GitHub também.
2. Use os algoritmos e classes da API Collections e da API Streams
3. Procura usar uma IDE (Eclipse, IntelliJ), mas usando o terminal pra compilar e executar. Java é muito "verboso", uma IDE ajuda na produtividade e tu tem ainda uma forma de entender bem as funções.

4 Problemas

Problema 1 UVa 1161

Problema 2 UVa 10424

Problema 3 UVa 119

Problema 4 UVa 11332

Problema 5 UVa 11942

Problema 6 UVa 12015

Problema 7 UVa 11044

Problema 8 UVa 414

Problema 9 UVa 10978

Problema 10 UVa 11093

Problema 11 UVa 11192

Problema 12 UVa 11496

Problema 13 UVa 10018

Problema 14 UVa 10945

Problema 15 UVa 156

Problema 16 UVa 195

Problema 17 UVa 454

Problema 18 UVa 10098

Problema 19 URI 1024

Problema 20 URI 1120