

Introdução à Maratona de Programação

Maratona de Programação FEI

Prof. Charles Ferreira
cferreira@fei.edu.br

*Introdução
Maratona
Programação* à
de

Maratona de
Programação

Objetivo

Treinar equipes para competir

- Treinar resolução de problemas
- Aprender sobre computação

Página do github (em construção)

- Material de apoio
- Informações compartilhadas

*Introdução
Maratona
Programação* à
de

Problemas da maratona
de programação

Sites de competição

Materiais de estudo e exercícios para aperfeiçoamento

Sites de competição

Materiais de estudo e exercícios para aperfeiçoamento

- beecrowd.com (Beecrowd)



Sites de competição

Materiais de estudo e exercícios para aperfeiçoamento

- beecrowd.com (Beecrowd)





- <https://usaco.guide/> (USACO)





Sites de competição

Materiais de estudo e exercícios para aperfeiçoamento

- beecrowd.com (Beecrowd) The Beecrowd logo consists of a stylized yellow and purple hexagonal shape above the word "beecrowd" in a purple, lowercase, sans-serif font.
- <https://usaco.guide/> (USACO) The USACO Guide logo features a blue circular icon with a white stylized 'A' shape inside, followed by the text "USACO Guide" in a white, sans-serif font, all on a dark blue rectangular background.
- <https://cses.fi/> (CSES) The CSES logo displays the letters "CSES" in a stylized, outlined, gold-colored font, set against a dark grey rectangular background.

Sites de competição

Materiais de estudo e exercícios para aperfeiçoamento

- beecrowd.com (Beecrowd) 
- <https://usaco.guide/> (USACO) 
- <https://cses.fi/> (CSES) 
- <https://youkn0wwho.academy/topic-list> (YouKnowWho Academy) 

Sites de competição

Materiais de estudo e exercícios para aperfeiçoamento

- beecrowd.com (Beecrowd) 
- <https://usaco.guide/> (USACO) 
- <https://cses.fi/> (CSES) 
- <https://youkn0wwho.academy/topic-list> (YouKnowWho Academy) 
- <https://codeforces.com/> (CodeForces) 

Exemplo: Problemas de competição

Basicamente um problema da maratona é constituído de três partes

- Enunciado (descrição do problema)
- Conjunto de Entradas
- Saída esperada

Enunciado: Exemplo

Problem A – Analyzing Contracts

Author: Agustín Santiago Gutiérrez, Argentina

Doctor Kruskal is starting a tiberium trading business. They have N possible suppliers of tiberium, and many clients interested in receiving tiberium to run their own industries.

Calendar days are numbered chronologically using positive integers, and each supplier is identified by a distinct integer from 1 to N . Supplier i can supply tiberium on any day from day S_i onwards, but not on the days strictly before S_i . They charge a price of P_i dollars per day for such a service. Since Kruskal is very smart, the list of suppliers contains only the best suppliers in the city. Besides, it is the case that $S_i < S_{i+1}$ and $P_i > P_{i+1}$ for $i = 1, 2, \dots, N - 1$.

Kruskal's system keeps a database of available clients. Initially, this database is empty and contains no clients. Clients will be arriving one by one, and each of them is immediately added to the database upon arrival. The j -th client is interested in receiving tiberium on any day up to day E_j inclusive. For each day that they receive tiberium, their industry will generate R_j dollars of gross revenue. Thus, if Kruskal matches supplier i to client j , the final profit of this whole operation after deducting the tiberium cost will be $(R_j - P_i) \times (E_j - S_i + 1)$, where $S_i \leq E_j$, as otherwise no tiberium could be provided.

At any moment, Kruskal's system can quickly compute, for any particular supplier i , the optimal client among those in the database, so that the profit of the operation when matching the supplier and the client is maximized, and it can report such maximum profit. It might be the case that a positive profit for a supplier cannot be achieved with any of the available clients; in that case, the system reports a profit of zero.

Notice that when Kruskal's system is requested to compute the maximum profit for a given supplier, that supplier is matched with at most one of the available clients, and in that case, such a match has no effect at all on future operations. This means that both the supplier and the client can be considered again for future matchings.

Your task is to implement Kruskal's system.

Costumam
ser longos

Importante saber
ler em inglês

Entrada: Exemplo

Input

The first line contains an integer N ($1 \leq N \leq 2 \times 10^5$) indicating the number of suppliers.

The i -th of the next N lines describes supplier i with two integers S_i and P_i ($1 \leq S_i, P_i \leq 10^9$), denoting respectively the start day and the price per day for the supplier. It is guaranteed that $S_i < S_{i+1}$ and $P_i > P_{i+1}$ for $i = 1, 2, \dots, N - 1$.

The next line contains an integer Q ($1 \leq Q \leq 2 \times 10^5$) representing the number of operations that must be processed. Operations are described in the next Q lines, in the order they are executed in the system, one operation per line. There are two types of operations.

If the operation adds a client to the database, the line contains the lowercase letter “c”, followed by two integers E and R ($1 \leq E, R \leq 10^9$), indicating respectively the end day and the gross revenue per day for the client.

If the operation requests to compute the maximum profit for a supplier, the line contains the lowercase letter “s”, followed by an integer I ($1 \leq I \leq N$) that identifies the supplier. It is guaranteed that the input contains at least one operation of this type.

Saída: Exemplo

Output

Output a line for each operation of type “s”. The line must contain an integer indicating the maximum possible profit when matching an available client with the given supplier. Write the results of the operations in the order they appear in the input.

Sample input 1	Sample output 1
4	0
2 8	18
4 5	35
7 3	28
9 2	16
11	84
s 1	16
c 10 10	28
s 1	108
s 2	
s 3	
s 4	
c 7 26	
s 2	
s 4	
s 3	
s 1	

*Introdução
Maratona
Programação* à
de

Linguagens de
Programação

Linguagens

Geralmente as linguagens mais comuns são:

- C++
- C
- Java
- Python
- Outras

Linguagens

Geralmente as linguagens mais comuns são:

- C++

Nossa escolha:
devido ao desempenho
e as bibliotecas (STL)

- C

- Java

- Python

- Outras

Linguagens

Geralmente as linguagens mais comuns são:

- C++

Nossa escolha:
devido ao desempenho
e as bibliotecas (STL)

- C

- Java

- Python

- Outras

Interessante ter conhecimentos sobre
alguma outra linguagem além de C++

Compilação e execução de programas

Habilidades que serão desenvolvidas e aperfeiçoadas

Compilação e execução de programas

Habilidades que serão desenvolvidas e aperfeiçoadas

- Linux (Sistema Operacional usado na competição)

Compilação e execução de programas

Habilidades que serão desenvolvidas e aperfeiçoadas

- Linux (Sistema Operacional usado na competição)
 - Utilizar o terminal
-



Compilação e execução de programas

Habilidades que serão desenvolvidas e aperfeiçoadas

- Linux (Sistema Operacional usado na competição)
 - Utilizar o terminal
-
- Compilar um programa pela linha de comando



Compilação e execução de programas

Habilidades que serão desenvolvidas e aperfeiçoadas

- Linux (Sistema Operacional usado na competição)
 - Utilizar o terminal
-
- Compilar um programa pela linha de comando
 - Executar um programa pela linha de comando



Compilação e execução de programas

Habilidades que serão desenvolvidas e aperfeiçoadas

- Linux (Sistema Operacional usado na competição)
 - Utilizar o terminal
-
- Compilar um programa pela linha de comando
 - Executar um programa pela linha de comando
 - Entender os erros do compilador



*Introdução à
Maratona de
Programação*

Editores de Texto

Editores de texto

Princípio importante:

Qualquer editor de texto lhe serve

É importante que o editor de texto seja um acelerador e não uma muleta!

Editores usados comumente

Nosso laboratório possui:

- VIM
- NeoVim
- Gedit
- VS Code

Vi IMproved

Principais características



Vi IMproved

Principais características

- Roda no terminal ✓



Vi IMproved

Principais características

- Roda no terminal ✓
- Keyboard Driven ✓



Vi IMproved

Principais características

- Roda no terminal ✓
- Keyboard Driven ✓
- Modular ✓



Vi IMproved



Principais características

- Roda no terminal ✓
- Keyboard Driven ✓
- Modular ✓
- Programável ✓

Vi IMproved



Principais características

- Roda no terminal ✓
- Keyboard Driven ✓
- Modular ✓
- Programável ✓
- Tem uma curva de aprendizado grande ✗

Vi IMproved



Principais características

- Roda no terminal ✓
- Keyboard Driven ✓
- Modular ✓
- Programável ✓
- Tem uma curva de aprendizado grande ✗
- Não é amigável (inicialmente) ✗

*Introdução
Maratona
Programação* à
de

Digitação

Aprender a digitar

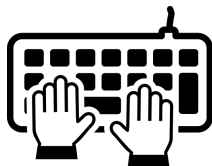
Importante fazer um curso de digitação

- Aprender a digitar com as duas mãos
- Posicionar as mãos corretamente
- Recomendação: typingclub.com (existem outros)

Exercício de digitação

Vamos fazer um pequeno experimento ...

- Abra um editor de texto qualquer (gedit, notepad, etc.)
- Posicione suas mãos no teclado como faria normalmente para digitar



Exercício de digitação

Tente digitar o seguinte texto sem olhar para o teclado

Exercício de digitação

Tente digitar o seguinte texto sem olhar para o teclado

```
1 | #include<iostream>
2 |
3 | using namespace std;
4 |
5 | int main(){
6 |     cout << "hello world\n";
7 |     return 0;
8 | }
```

*Introdução à
Maratona de
Programação*

Rotina de estudos

Rotina de estudos

Importante criar uma rotina de estudos

- Equilíbrio entre resolver exercícios e aprender novos conceitos

Consistência vence intensidade!

- Melhor treinar pouco frequentemente do que muito em um único dia.

Sugestão de treinamento inicial

Planilha de controle de estudos

- Link: [Google Sheets](#)
- Solicite a adição de uma aba com o seu nome
- Va marcando os exercícios que já conseguiu resolver ✓

ONDE	TEMA	LVL	LINK	bernardo
FEI	01 - Programação Básica	1	Bello World!	<input type="checkbox"/>
FEI	01 - Programação Básica	1	Extremamente Básico	<input type="checkbox"/>
FEI	01 - Programação Básica	1	Área do Círculo	<input type="checkbox"/>
FEI	01 - Programação Básica	1	Soma Simples	<input type="checkbox"/>
FEI	01 - Programação Básica	1	Produto Simples	<input type="checkbox"/>
FEI	01 - Programação Básica	1	Média 1	<input type="checkbox"/>
FEI	01 - Programação Básica	1	Média 2	<input type="checkbox"/>
FEI	01 - Programação Básica	1	Diferença	<input type="checkbox"/>
FEI	01 - Programação Básica	1	Salário	<input type="checkbox"/>
FEI	01 - Programação Básica	1	Salário com Bônus	<input type="checkbox"/>
FEI	01 - Programação Básica	1	Consumo	<input type="checkbox"/>
FEI	01 - Programação Básica	1	Distância	<input type="checkbox"/>
FEI	01 - Programação Básica	1	Conversão de Tempo	<input type="checkbox"/>
FEI	01 - Programação Básica	1	O Maior	<input type="checkbox"/>
FEI	01 - Programação Básica	1	Área	<input type="checkbox"/>
FEI	01 - Programação Básica	1	Esfera	<input type="checkbox"/>
FEI	01 - Programação Básica	1	Cálculo Simples	<input type="checkbox"/>
FEI	01 - Programação Básica	1	Cédulas	<input type="checkbox"/>
FEI	01 - Programação Básica	1	Custo de Combustível	<input type="checkbox"/>
FEI	01 - Programação Básica	1	Idade em Dias	<input type="checkbox"/>

Sugestão de treinamento inicial

Ao travar por muito tempo em algum problema:



Sugestão de treinamento inicial

Ao travar por muito tempo em algum problema:

- Tente conversar com quem já resolveu.



Sugestão de treinamento inicial

Ao travar por muito tempo em algum problema:

- Tente conversar com quem já resolveu.
- Procure entender a descrição da solução.



Sugestão de treinamento inicial

Ao travar por muito tempo em algum problema:

- Tente conversar com quem já resolveu.
- Procure entender a descrição da solução.
- Rabisque em um papel ou na lousa.



Sugestão de treinamento inicial

Ao travar por muito tempo em algum problema:

- Tente conversar com quem já resolveu.
- Procure entender a descrição da solução.
- Rabisque em um papel ou na lousa.
- Por fim, tente efetivamente codificar.



*Introdução
Maratona
Programação* à
de

Encontros (conteúdo
programático)

Encontros

Sala da Maratona: K409



- Horário de treinamento:

-
- Segunda - Sexta: 13:00 às 18:00
 - Responsáveis: Gabriel e Alexandre

-
- Segunda - Sexta: 09:30 às 12:00
 - Responsáveis: Felipe, Bernardo e Rafael

Cronograma

Cronograma

Data	Assunto	Palestrante
08/04/25	Introdução à maratona de programação	Charles
15/04/25	Introdução à C++	Gabriel
22/04/25	Análise de Algoritmos	Charles
29/04/25	Busca Binária	Bernardo
06/05/25	STL	Gabriel e Bernardo
13/05/25	Grafos 1	Bernardo
20/05/25	Grafos 2	à definir
27/05/25	Algoritmos Gulosos	Charles
03/06/25	Programação Dinâmica	Charles
10/06/25	SegTree	à definir
Recesso		
29/08/25	Matemática para Maratona	Alexandre
12/09/25	Soma de prefixos / scripts	à definir

Obrigado

cferreira@fei.edu.br