

Aula Prática - Algoritmos de Ordenação

Clausius G. Reis

Leandro A. Amaral

Tiago S. Nazaré

Vanessa Q. Marinho

22 de Junho de 2015

Exercício 1: Grid de largada (Individual)

Retirado da Maratona de Programação da SBC - ACM ICPC - 2012

Na Nlogônia, vai ser realizada a sensacional final mundial da fórmula 17. Os competidores se alinham na largada e disputam a corrida. Você vai ter acesso aos grids de largada e de chegada. A questão é determinar o número mínimo de ultrapassagens que foram efetuadas durante a competição.

Entrada:

Cada caso de teste utiliza três linhas. A primeira linha de um caso de teste contém um inteiro N indicando o número de competidores. Cada competidor é identificado com um número de 1 a N . A segunda linha de cada caso tem os N competidores, em ordem do grid de largada. A terceira linha de cada caso tem os mesmos competidores, porém agora na ordem de chegada.

Saída:

Para cada caso de teste imprima uma linha contendo um único número inteiro, que indica o número mínimo de ultrapassagens necessárias para se chegar do grid de largada ao grid de chegada.

Restrições:

- $2 \leq N \leq 24$

Casos de teste:

5	
1 2 3 4 5	
3 1 2 5 4	
	3
5	
3 1 2 5 4	3
1 2 3 4 5	
	4
5	
3 1 2 5 4	
5 3 2 1 4	

Exercício 2: Frequência na aula (Individual)

Retirado da Olimpíada Brasileira de Informática - OBI 2012

Certa vez, numa aula, a professora passou um filme para os alunos assistirem. Durante este filme, ela passou uma lista de presença em sua sala para verificar a presença dos alunos, onde cada aluno deveria inserir apenas seu número de registro. Alguns alunos contudo, como possuem amigos que fogem da aula, decidiram ser camaradas e inseriram os números de registro de seus amigos fujões. O problema é que muitos alunos são amigos de alunos que fogem da aula e alguns números de registro acabaram sendo repetidamente inseridos na lista de presença. Além de tudo, alguns dos alunos que se esperava que não estivessem na aula de fato estavam!

A professora, ao notar que a lista de presença continha alguns números repetidos, ficou sem entender, mas decidiu dar um voto de confiança e dar presença a todos os alunos cujos números de registro estavam na lista. Como são muitos alunos na sala e muitos números com repetição, ela pediu a sua ajuda para determinar o total de alunos que receberam presença na aula.

Entrada:

A primeira linha da entrada contém um número inteiro N , que informa a quantidade de números de registro que apareceram na lista de presença. Cada uma das N linhas seguintes contém um número de registro V_i que foi inserido na lista de presença.

Saída:

Seu programa deve imprimir uma única linha, contendo apenas um número inteiro, o número de alunos que receberam presença.

Restrições:

- $1 \leq N \leq 10^5$
- Para cada elemento V_i , $0 \leq V_i \leq 10^6$

Informações sobre pontuação:

- Em um conjunto de casos que totaliza 40 pontos, $N \leq 10^3$ e $V_i \leq 10^3$

Casos de teste:

3	
2	
3	3
1	

Exercício 3: Olimpíadas (Grupo)

Olimpíada Brasileira de Informática – OBI 2009

O Comitê Olímpico Internacional (COI) está visitando as cidades candidatas a sediar as Olimpíadas de 2016. O Rio de Janeiro é uma das cidades concorrentes, mas a competição é muito acirrada.

O COI tem um conjunto de exigências que devem ser obedecidas pelas cidades candidatas, como boas arenas para os jogos (ginásios, campos de futebol, pistas de atletismo, parque aquático,...), bons alojamentos, um plano para o tráfego de veículos durante os jogos, etc. Durante sua visita ao Rio de Janeiro, o COI colocou ainda mais uma exigência: a demonstração da qualidade dos sistemas de informática. Especificamente, o COI quer que a organização local demonstre a sua capacidade em informática produzindo um programa que gere a classificação final dos países, considerando o número de medalhas recebidas pelos atletas de cada país.

Tarefa:

Sua tarefa é escrever um programa que, dada a informação dos países que receberam medalhas de ouro, prata e bronze em cada modalidade, gere a lista de classificação dos países na competição. Nesta tarefa, os países serão identificados por números inteiros. O melhor colocado deve ser o país que conseguiu o maior número de medalhas de ouro. Se houver empate entre países no número de medalhas de ouro, o melhor colocado entre esses é o país que conseguiu o maior número de medalhas de prata. Se houver empate também no número de medalhas de prata, o melhor colocado entre esses é o país que recebeu o maior número de medalhas de bronze. Se ainda assim houver empate entre dois países, o melhor classificado é o que tem o menor número de identificação.

Entrada:

A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (Normalmente o teclado).

A primeira linha da entrada contém dois números N e M , separados por um espaço em branco, indicando respectivamente o número de países ($1 \leq N \leq 100$) e o número de modalidades envolvidas na competição ($1 \leq M \leq 100$). Os países são identificados por um número inteiro de 1 a N .

Cada uma das M linhas seguintes contém três números inteiros O , P e B , separados por um espaço em branco, representando os países cujos atletas receberam respectivamente medalhas de ouro ($1 \leq O \leq N$), prata ($1 \leq P \leq N$) e bronze ($1 \leq B \leq N$). Assim, se uma das M linhas contém os números 3 2 1, significa que nessa modalidade a medalha de ouro foi ganha pelo país 3, a de prata pelo país 2 e a de bronze pelo país 1.

Saída:

Seu programa deve imprimir, na saída padrão, uma linha contendo N números, separados por um espaço em branco, representando os países na ordem decrescente de classificação (o primeiro número representa o país que é o primeiro colocado, o segundo número representa o país que é o segundo colocado, e assim por diante).

Informações sobre pontuação:

- Em um conjunto de casos que totaliza 30 pontos, $N \leq 10$ e $M \leq 10$
- Em um conjunto de casos que totaliza 70 pontos, $N \leq 50$ e $M \leq 50$

Casos de teste:

2 2 2 1 2 1 2 2	2 1
-----------------------	-----

4 3 3 2 1 4 3 1 4 3 1	4 3 2 1
--------------------------------	---------

3 3 3 1 2 2 3 1 1 2 3	1 2 3
--------------------------------	-------