

MC458 Projeto e Análise de Algoritmos I - 2016/02

Laboratório 2

Docente: Cid Carvalho de Souza

Monitora PED: Amanda Resende

Monitor PAD: Allan Sapucaia Barboza

Descrição

A América Latina Logística (ALL) é uma empresa que opera um grande terminal ferroviário onde organiza comboios de vagões que são rebocados pelas locomotivas da Companhia Ferroviária (CF). A organização dos vagões que são acoplados a uma locomotiva é uma tarefa complexa e bastante custosa. Isso porque a ordem em que eles são acoplados à locomotiva deve ser exatamente a ordem inversa em que eles são deixados ao longo da viagem feita pelo trem. Assim, o último vagão é o primeiro a ser desacoplado do comboio, o penúltimo é o segundo, e assim por diante, até que o primeiro vagão seja o último a ser desacoplado, restando apenas a locomotiva.

Os vagões são entregues no terminal da ALL pela empresa Vale. Os N **vagões** que irão compor um comboio são identificados por **rótulos** $R(1), \dots, R(N)$, dados por números inteiros não negativos **todos diferentes entre si**, de modo que o I -ésimo vagão na ordem entregue pela Vale tem rótulo dado por $R(I)$. Os significados destes rótulos não vêm ao caso, mas é importante saber que eles estão associados à ordem de desacoplamento do comboio. Assim, andando ao longo do trem que deve sair do terminal da ALL a partir da locomotiva, os rótulos dos vagões deveriam aparecer na ordem crescente dos rótulos.

Ocorre que a Vale, contrariamente à ALL, tem pouco espaço para fazer manobras com os vagões e, para não atrasar as operações, acaba entregando os vagões para a ALL em uma ordem arbitrária, mesmo incorrendo em uma **Multa Contratual (MC)**.

A multa contratual é computada da seguinte forma. Primeiramente calcula-se o total de pares de vagões que estão em ordem trocada, chamado de **OT**. Note que um par (I, J) , onde $1 \leq I < J \leq N$, estará *trocado* na ordem dos vagões entregues pela Vale se $R(I) > R(J)$. Em seguida, para todo vagão $I \in \{1, \dots, N\}$, sendo $P(I)$ a posição correta do vagão I na entrega feita pela Vale para a ALL, calcula-se $V(I) = |P(I) - I|$, que pode ser entendido como a medida de quão distante o vagão I está da sua posição correta $P(I)$. Com isto, define-se $\mathbf{M} = \max_{I=1, \dots, N} V(I)$ e chega-se ao valor da multa contratual dada por: **MC** = $\mathbf{M} \times \mathbf{OT}$.

Em outras palavras, a MC é dada pelo total de troca de pares de vagões (**OT**) multiplicado pela maior distância entre a posição de entrega de um vagão pela Vale e a sua posição correta no comboio preparado pela ALL (M). Por exemplo, se a Vale entrega 4 vagões para a ALL ($N = 4$) na seguinte ordem 10, 30, 40 e 20, tem-se que $OT = 2$ ((2, 4) e (3, 4)); $V(1) = 0$, $V(2) = 1$, $V(3) = 1$ e $V(4) = 2$; $M = V(4) = 2$; e, finalmente $MC = 2 \times 2 = 4$.

Como a leitura e a comparação dos rótulos nos vagões é um processo sujeito a erros, a ALL quer fazer poucas comparações de rótulos para calcular o valor de *OT*. Na verdade, conhecido N , os engenheiros da ALL calculam um valor C , que é o **número máximo** destas comparações para o qual consideram o processo confiável.

Neste laboratório, você deverá ajudar a ALL a calcular o **valor da multa contratual**, atendendo à exigência dos engenheiros de não realizar mais do que C comparações de rótulos entre vagões.

Entrada

O arquivo de entrada consiste em 2 linhas. A primeira contém um inteiro positivo N entre 1 e 1000000, que representa a quantidade de vagões que irão compor o comboio, e um inteiro positivo C , que é a quantidade máxima de comparações permitidas entre os rótulos. Já a segunda linha contém N números inteiros não negativos, entre 0 e $10N$, que representam o valor do rótulo de cada vagão.

Saída

A saída do programa é composta por 2 linhas. A primeira contém um inteiro não negativo MC que corresponde ao valor da multa contratual que a Vale deveria pagar à ALL. A segunda é composta pela *string* *OK*, se o número de comparações entre os rótulos foi menor ou igual ao máximo permitido C , ou uma mensagem de erro caso contrário.

Exemplo 1

```
4 16          /* entrada */
1 2 3 10      /* entrada */
0             /* saída */
OK            /* saída */
```

Exemplo 2

```
4 16          /* entrada */
10 2 3 1      /* entrada */
15            /* saída */
OK            /* saída */
```

Exemplo 3

```
4 12          /* entrada */
10 3 2 1      /* entrada */
18            /* saída */
OK            /* saída */
```

Especificações

O programa deve ser implementado em C++. Nesse projeto é permitido o uso de quaisquer estruturas de dados ou rotinas de uma biblioteca do C++.

No Susy serão fornecidos 4 arquivos: *main.cpp*, *pint.cpp*, *pint.hpp* e *solution.hpp*. O arquivo *main.cpp* lida com as entradas e saídas dos dados e é responsável por inicializar a execução do laboratório e verificar o número de comparações realizadas entre rótulos com o valor máximo C . A classe *Pint* é responsável por representar inteiros e tem apenas duas operações: comparação e soma (ver o arquivos *pint.hpp* e *pint.cpp*). Ela é responsável por contar o número de comparações realizadas entre os rótulos. Por fim, a função *solution* (*solution.hpp*) tem como entrada um vetor de *Pint* com N elementos e contém exemplos de como as operações de comparação e soma são realizadas.

Os arquivos *main.cpp*, *pint.cpp*, *pint.hpp*, **não precisam e não devem ser alterados**. Você deve apenas implementar a função *solution* que está dentro do arquivo *solution.hpp*, portanto, o **único arquivo** que deverá ser alterado e submetido no Susy será o *solution.hpp*. Podem ser criadas novas funções auxiliares (além da *solution*) no arquivo *solution.hpp* desde que elas não infrinjam nenhuma das regras. Todos os arquivos estão disponíveis na aba “Arquivos auxiliares” do Susy, incluindo o *Makefile*.

Avaliação

Haverá 20 testes abertos e 20 testes fechados. A nota do projeto é proporcional ao número de teste bem-sucedidos. Mais precisamente, se T é o número de testes bem-sucedidos do seu programa, então sua nota será $(\frac{T}{40}) \times 10 = \frac{T}{4}$.

Prazo de submissão

O programa pode ser submetido até a seguinte data:

- 08:00h de 20 de outubro (quinta-feira).

Observações

- O número máximo de submissões é 10.
- Para a avaliação será considerada apenas a última versão do programa submetido.
- Para submissão no Susy utilize o número do seu RA em **Usuário** e a senha utilizada na DAC em **Senha**.
- Para manipular os objetos do tipo *Pint* você deve usar **exclusivamente** os procedimentos disponibilizados nos arquivos *pint.hpp* e *pint.cpp*. **O uso de outra forma de manipulação deste tipo de dados que não seja via os procedimentos que são disponibilizados no enunciado será considerado como uma fraude e, portanto, levará à aplicação estrita das regras previstas para este caso.**