

MC458 Projeto e Análise de Algoritmos I – 2016/02

Laboratório 1

Docente: Cid Carvalho de Souza

Monitora PED: Amanda Resende

Monitor PAD: Allan Sapucaia Barboza

Descrição

Chuck e Casey fazem parte de uma equipe de espões que trabalham para a CIA/NSA, que também conta com Sarah. Eles vivem em Burbank e trabalham na Buy More como disfarces. Casey é um pessoa super competitiva e odeia ser superado em qualquer coisa, principalmente se for para o Chuck. Entre uma missão e outra, Chuck adora provocar Casey e sempre o desafia para algum jogo.

O jogo dessa vez consiste em resolver o seguinte problema. São dadas N embalagens identificadas com rótulos de 0 a $N-1$, completamente lacradas, com exatamente a mesma aparência e preenchidas com bolinhas de gude por Sarah. Sabe-se que todas as embalagens possuem o mesmo peso, exceto **uma única embalagem** que tem **peso menor** que as demais. Além disso, disponibiliza-se uma balança de dois pratos (idênticos), no qual supõe-se poder ser colocada para pesagem qualquer subconjunto das N embalagens. Assim, pode-se fazer a pesagem de dois destes subconjuntos na balança para comparar o peso dos mesmos.

O objetivo principal do jogo é, usando a balança fornecida, encontrar a embalagem que é mais leve que as demais. Perceba que se não for estabelecido um limite no número de pesagens que podem ser feitas com auxílio da balança, o problema é trivialmente resolvido. Por isso, na entrada do problema, é dado também um valor inteiro positivo M que limita o número **máximo** de pesagens que podem ser efetuadas para se chegar à solução.

Neste laboratório você deve projetar um algoritmo para resolver o problema e ajudar Casey a derrotar Chuck. Para isso, você deverá implementar a função *solution* no arquivo *solution.hpp*.

Nota: em todas as instâncias de teste é garantido que haverá exatamente uma embalagem com peso diferente e menor do que o peso das demais.

Entrada

O arquivo de entrada consiste em 2 linhas. A primeira contém um inteiro N , que representa a quantidade de embalagens preparadas por Sarah e um inteiro M , que é a quantidade máxima de pesagens que podem ser realizadas. A segunda linha contém N números inteiros, que representam o peso de cada embalagem, onde $N - 1$ pesos são iguais e 1 peso é menor que os demais.

Saída

A saída consiste em duas linhas. A primeira contém o rótulo da embalagem que possui o menor peso (valor entre 0 e $N - 1$). A segunda é composta pela *string OK*, se o número de pesagens foi menor ou igual ao máximo permitido M , ou uma mensagem de erro caso contrário.

Exemplo 1

```
12 5                /* entrada */
2 2 2 1 2 2 2 2 2 2 /* entrada */
3                  /* saída */
OK                 /* saída */
```

Exemplo 2

```
12 5                /* entrada */
2 2 2 2 2 2 2 2 2 1 2 2 /* entrada */
9                  /* saída */
Numero excessivo de comparacoes /* saída */
```

Especificações

O programa deve ser implementado em C++. Neste 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. 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 pesagens realizadas e comparar com o valor máximo M , que também faz parte da instância. 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*. 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 testes 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 27 de setembro (terça-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 para acessar o email da 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.**