

MC458 Projeto e Análise de Algoritmos I - 2016/02

Laboratório 3

Docente: Cid Carvalho de Souza

Monitora PED: Amanda Resende

Monitor PAD: Allan Sapucaia Barboza

Descrição

A Universidade Caipira (UNICAIP) lançou um programa de inclusão para acesso de alunos do ensino público aos cursos de graduação. No total são M vagas disponibilizadas pela universidade para o programa. O processo seletivo funciona do seguinte modo.

Primeiramente a UNICAIP seleciona K escolas públicas que têm pelo menos N alunos no 3º ano do ensino médio para participar do programa. Cada uma destas K escolas envia para a universidade uma lista com o identificador e a nota dos seus N alunos melhores classificados no ENEM (tanto o identificador quanto a nota são números inteiros não negativos). Lembre-se que, evidentemente, quanto maior for a nota de um aluno no ENEM, melhor será a sua classificação no Exame. Outra informação relevante é que, na prática, as listas recebidas pela UNICAIP não possuem nenhuma ordenação a priori.

A partir destes dados, as escolas são ordenadas, em ordem crescente, de acordo com a média obtida pelos seus N alunos. Note-se que as notas dos alunos no ENEM são sempre dadas por números inteiros não negativos e que as médias das escolas são sempre computadas desprezando-se as casas decimais (note que isto é feito não por arredondamento, mas sim por truncamento).

Finalmente, sendo $\{E_1, \dots, E_K\}$ a lista ordenada das escolas participantes do programa (de acordo com as médias), as M vagas disponibilizadas pela UNICAIP para o programa são preenchidas de acordo com a regra apresentada abaixo.

```
1   $c \leftarrow 1$                                 /* posição do aluno no processo seletivo da UNICAIP */
2   $i \leftarrow 1$                                 /* classificação de um aluno em uma escola */
3   $j \leftarrow 1$                                 /* índice da escola */
4  enquanto  $c \leq M$  faça                        /* preenche todas vagas */
5      Chamar aluno da escola  $E_j$  com a  $i$ -ésima melhor nota no ENEM na
         $c$ -ésima posição do processo seletivo da UNICAIP
6       $j++$                                         /* próxima escola */
7       $c++$                                         /* próxima posição do processo seletivo */
8      se  $j = K$  então                            /* chamou o  $i$ -ésimo melhor de todas escolas */
9           $i \leftarrow i++$                         /* considera o próximo aluno na ordem */
10          $j \leftarrow 1$                             /* volta para a primeira (melhor) escola */
11     fim
12 fim
```

Em outras palavras, a regra é classificar os alunos das escolas agrupando-os em

blocos de K em K alunos, cada um deles contendo um aluno de cada escola, exceto possivelmente o último bloco que poderá ter menos de K alunos. Em cada bloco, a ordem de classificação no processo seletivo sempre respeita a ordem relativa das escolas segundo a média dos seus alunos no ENEM. Assim, de um modo geral, o i -ésimo aluno melhor classificado da escola E_j será classificado na posição $(i - 1) \times K + j$ do processo seletivo.

Perceba que casos de empate entre as médias das escolas são sempre resolvidos de modo que a ordem original dos dados de entrada sejam respeitadas. Assim, por exemplo, se o cálculo da média das notas dos N alunos de duas escolas E_i e E_j levam a resultados idênticos, a escola E_i será privilegiada em relação à escola E_j sempre que $i < j$ (sendo estes os índices na entrada). Vale ressaltar que é garantido que os identificadores são únicos e que os alunos de uma mesma escola tem notas distintas.

Considere o seguinte exemplo: Seja $K = 3$, $M = 8$ e $N = 4$. Assim, tem-se 3 escolas com pelos menos 4 alunos e a UNICAIP está oferecendo 8 vagas.

- As escolas disponibilizaram os seus dados para a universidade como listas de pares da forma (identificador_aluno, nota) como se segue:
 - $E1 : (11, 8)(21, 10)(31, 7)(41, 9)$
 - $E2 : (12, 8)(22, 10)(32, 9)(42, 13)$
 - $E3 : (13, 6)(23, 8)(33, 12)(43, 16)$
- As médias (truncadas) das escolas são
 - $E1 = (8 + 10 + 7 + 9)/4 = 8$
 - $E2 = (8 + 10 + 9 + 13)/4 = 10$
 - $E3 = (6 + 8 + 12 + 16)/4 = 10$ (Nota: divisão inteira)
- Ordem das escolas: ($E2, E3, E1$).
- Ordem dos alunos nas escolas:
 - $E1 : (21, 10)(41, 9)(11, 8)(31, 7)$
 - $E2 : (42, 13)(22, 10)(32, 9)(12, 8)$
 - $E3 : (43, 16)(33, 12)(23, 8)(13, 6)$
- Alunos chamados ordenados de acordo com as regras do processo seletivo:

$(42, 13)(43, 16)(21, 10)(22, 10)(33, 12)(41, 9)(32, 9)(23, 8)$

Neste laboratório, você deverá ajudar a UNICAIP a encontrar a lista ordenada dos alunos aprovados no processo seletivo segundo a regra descrita anteriormente. No entanto, isso deverá ser feito de modo que o número máximo de comparações entre notas (ou médias de notas) seja no máximo C . Perceba então que, para o seu programa fornecer a resposta correta para todas as instâncias, o algoritmo que ele implementa deverá ser projetado de modo a fazer no máximo C comparações.

Entrada

O arquivo de entrada consiste em $K + 1$ linhas. A primeira contém 4 inteiros positivos K (quantidade de escolas), M (quantidade de vagas disponibilizadas pela UNICAIP), N (quantidade mínima de alunos que a escola deve ter) e C (número máximo de comparações), onde $K \times N > 10M$, $N > 4$ e $M > K$. Já as demais K linhas são compostas por N pares de números inteiros não negativos da forma “identificador_aluno nota”.

Saída

A saída do programa é composta por 2 linhas. A primeira contém M pares de inteiros não negativos da forma “identificador_aluno nota”, que corresponde aos M alunos selecionados pela UNICAIP na ordem estabelecida pela regra descrita no enunciado. A segunda é composta pela *string* *OK*, se o número de comparações entre os valores de notas e médias for menor ou igual ao máximo permitido C , ou uma mensagem de erro caso contrário.

Exemplo 1 (Nota: este exemplo não respeita a restrição $K \times N > 10M$ e foi adicionado aos testes apenas para facilitar o entendimento do problema.)

```
3 8 4 25          /* entrada */
21 11 11 8 41 7 31 6    /* entrada */
42 13 12 12 22 8 32 7   /* entrada */
43 16 33 12 23 8 13 6   /* entrada */
42 13 43 16 21 11 12 12 33 12 11 8 22 8 23 8 /* saída */
OK                  /* saída */
```

Exemplo 2 (Nota: este exemplo não respeita a restrição $K \times N > 10M$ e foi adicionado aos testes apenas para facilitar entendimento do problema.)

```
3 8 4 25          /* entrada */
12 12 42 13 32 7 22 8 /* entrada */
13 6 43 16 23 8 33 12 /* entrada */
11 8 21 11 31 6 41 7  /* entrada */
42 13 43 16 21 11 12 12 33 12 11 8 22 8 23 8 /* saída */
OK                  /* saída */
```

Especificações

O programa deve ser implementado em C++. **Neste trabalho está explicitamente proibido o uso de bibliotecas adicionais do C++ e a função *qsort* do C, assim como outras funções de ordenação e estruturas de dados complexas.**

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 com o valor máximo C . A classe *Pint* é responsável por representar inteiros e tem três operações: comparação, soma e divisão (ver o arquivos *pint.hpp* e *pint.cpp*). Ela é responsável por contar o número de comparações realizadas entre os valores de notas e médias. Por fim, a função *solution* (*solution.hpp*) tem como entrada um vetor de pares de *Pint* com N pares (identificador_aluno nota) e contém exemplos de como as operações de comparação, soma e divisão 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 02 de novembro (quarta-feira), portanto, em um prazo de **48 horas** a contar da divulgação do enunciado.

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 á aplicação estrita das regras previstas para este caso.**