

Trabalho Prático 3

Algoritmos e Estruturas de Dados III

30 de Novembro de 2018

Entrega: 09/12/2018 até 23:55.

~~Entrega: 04/12/2018~~

1 Descrição

A cidade de “Fabulouston” tem um ilustre morador chamado “**João Colotal**”. João é símbolo de simplicidade e humildade, mesmo morando em um cidade tão movimentada ele nunca se preocupa suas vestimentas.

Recentemente, participando de um evento beneficente do “Instituto Programar” ele aprendeu sobre localidade de referência e memória secundária. Mesmo sendo um conceito tão complexo, ele sabe que este conteúdo pode ser útil para melhorar a eficiência de diversos algoritmos.

Colotal ficou deslumbrado com tudo que aprendeu durante o evento, agora ele não se cansa de programar. Sempre que ele tem um tempo, ele pesquisa por vários problemas e operações importantes para serem feitas em blocos de memórias alocados sequencialmente. Um de seus preceptores está lecionando um curso online e pediu ao João Colotal para desenvolver um programa para calcular a média e mediana. O programa de João precisa então receber o número de Usuários e o número de cursos que o seu preceptor vai aplicar durante o ano. O problema é que as notas nessa modalidade dos cursos é dada por um número inteiro de 32 *bits*.

A tarefa principal do trabalho é conseguir calcular a média e mediana de uma matriz M para João Colotal, além disso apresentar também quais são os valores de média e mediana dos usuários. Só assim ele saberá que é possível resolver este problema de forma eficiente.

2 Definição

Neste trabalho, nós estamos interessados em encontrar a média aritmética, mediana de várias linhas uma matriz $N \times M$, além disso encontrar a média geral da matriz. Na Seção 2.1 temos a descrição do cálculo da média aritmética. Já na Seção 2.2 temos a descrição de como é feito o cálculo da mediana.

2.1 Média Aritmética

A Média Aritmética é definida como “a soma de todos os números em uma lista, dividido pelo número de itens naquela lista”. Por exemplo, a média aritmética dos números 2, 3 e 7 é 4, uma vez que $2 + 3 + 7 = 12$ e $12 \div 3 = 4$.

A Equação 1 apresenta a formula da média aritmética:

$$\bar{x} = \frac{1}{N} \times \sum_{i=1}^N x_i \quad (1)$$

Onde, \bar{x} representa a média obtida, N é o número de itens presentes e x_i é o valor individual de cada item na lista.

2.2 Mediana

A mediana é o número que está localizado na posição central de um conjunto de números, quando o conjunto está ordenado sequencialmente em ordem crescente. Por exemplo, a mediana do conjunto 2, 4, 10 é 4 e a mediana do conjunto -200, 2, 4, 100, 150 é, também 4.

Como calcular a mediana:

1. Ordene os números em ordem crescente.
2. Caso o tamanho do conjunto seja ímpar, a mediana é o número de posição $N/2$
3. Caso o tamanho do conjunto seja par, a mediana é a média entre os dois números centrais.

2.3 Objetivos

O objetivo desse trabalho prático é o desenvolvimento de um programas utilizando as técnicas e os conceitos sobre **localidade de referência** e **ordenação externa**.

Identifique os *tradeoffs* entre velocidade de execução e gasto de memória para calcular os seguintes indicadores:

- Calcular Média Aritmética por linha
- Calcular Mediana por linha
- Calcular a Média Aritmética Geral de toda a matriz

Voc

3 Entrada

A primeira linha do arquivo de entrada contém 2 números inteiros referentes ao número de usuários U ($1 \leq U \leq 3 \cdot 10^3$) e cursos C ($1 \leq C \leq 3 \cdot 10^3$). As próximas linhas contém uma sequência de C valores para cada usuário. No total temos $U \times C$ valores. Cada valor da sequência é um inteiro positivo e o seu valor máximo é 4,294,967,296. Valor de representação de um inteiro (*int*) de 32 *bits*.

Um exemplo de entrada para o programa está na Tabela 1.

Tabela 1: Exemplo *Toy* para ilustrar a entrada do problema. Esse teste está disponível em um arquivo no moodle.

input
4 4
3 5 9 10
7 1 2 4
12 8 20 13
15 6 17 19

Os alunos devem elaborar novos casos de testes. Os novos casos de testes podem ser compartilhados uns com os outros.

4 Saída

Para cada linha da matriz de entrada, deve ser impresso na saída dois valores separados por espaço vírgula . Do seguinte modo:

1º valor:

Média Aritmética da linha i

2º valor:

Mediana da linha i

Por fim, apresentar a média geral da matriz

3º valor:

Média Aritmética Geral (de toda a matriz)

Não pode existir espaço entre a resposta “Média,Mediana”. Outro aspecto importante é que os valores decimais devem ser impressos com apenas duas casas decimais. Um exemplo de saída para o programa está na Tabela 2.

Observação: Fique atento quanto a complexidade do seu algoritmo, avalie o *tradeoff* do seu algoritmo em relação a memória utilizada e a velocidade de execução do seu programa. Além disso, utilize o **diff** do Linux para conferir se o seu algoritmo está com a saída correta e de acordo com as respostas de exemplo.

Tabela 2: Exemplo *Toy* para ilustrar a saída do problema. Esse teste está disponível em um arquivo no moodle.

output
6.75,7
3.5,3
13.25,12.5
14.25,16
9.34

5 Limite de memória

Um dos desafios desse trabalho consiste em utilizar um tamanho de memória limitado para realizar a ordenação de um arquivo que não pode ser carregado por completo na mesma. Um dos parâmetros para execução do seu programa será o limite de memória que ele poderá utilizar. Esses limites devem ser respeitados. Existem ferramentas para medição do uso de memória, ou seja, controlar a quantidade de memória alocada para determinado programa durante sua execução. Por exemplo, *timeout* (visite o site: <https://github.com/pshved/timeout>) e *ulimit* (visite o site: <https://ss64.com/bash/ulimit.html>).

O limite de memória a ser considerado no programa será passado na chamada do executável, por exemplo:

```
./tp3 input.txt output.txt LIM_M
```

(2)

Onde `LIM_M` é o máximo de memória que seu algoritmo pode alocar. Esse valor será em MBs, ou seja, caso o valor de `LIM_M` seja 256, significa que seu programa poderá alocar somente 256Mb de memória.

6 Entrega

- A data de entrega desse trabalho é até às 23:55 do dia **02/12/2018** (04 de Dezembro de 2018).
- A penalização por atraso obedece à formula apresentada na Equação 3, onde d são os dias úteis de atraso na entrega do projeto.

$$\frac{2^{d-1}}{0.32\%}$$
(3)

- Submeta apenas um arquivo chamado `<número matricula>_<nome>.zip`. Não utilize espaços no nome do arquivo. Ao invés disso utilize o caractere “_”. Não envie o arquivo compactado .rar, o formato de arquivo ZIP é mais acessível que o RAR.

Arquivos fora deste formato não serão aceitos.

Exemplo: **2017648535_Joao_pe_de_feijao.zip**

- Não inclua arquivos compilados ou gerados por IDEs. **Apenas** os arquivos abaixo devem estar presentes no arquivo zip:
 - Makefile;
 - Arquivos fonte (*.c e *.h);
 - documentacao.pdf.
- Não inclua **nenhuma** pasta. Coloque todos os arquivos na raiz do zip.
- Siga rigorosamente o formato do arquivo de saída descrito na especificação. Tome cuidado com *whitespaces* e formatação dos dados de saída.
- **NÃO SERÁ NECESSÁRIO ENTREGAR DOCUMENTAÇÃO IMPRESSA!**
- A sua nota final será 0 se uma das partes (**documentação ou execução**) não for apresentada.

7 Documentação

A documentação não deve exceder o limite de **10 páginas**, sendo submetida no formato PDF e deve conter pelo menos os seguintes itens:

- Uma introdução do problema em questão com uma explicação clara e objetiva;
- Modelagem e solução proposta para o problema. O algoritmo deve ser explicado de forma clara. Se necessário, utilize pseudo-código e/ou esquemas ilustrativos.
- Análise de complexidade de tempo e espaço da solução usando o formalismo da notação assintótica visto em sala de aula;
- Análise experimental, variando-se o tamanho da entrada e quaisquer outros parâmetros que afetem significativamente a execução do algoritmo;
 - **Cabe a você gerar as entradas para esses experimentos.**
- Especificação da máquina utilizada nos experimentos realizados;
- Uma breve conclusão do trabalho implementado.

Um tutorial para elaboração da documentação e também um exemplo são apresentados no *minha.ufmg*¹. Para visualizar o tutorial para elaboração da documentação (**clique aqui**), já para visualizar o exemplo de uma documentação pronta (**clique aqui**).

¹<https://sistemas.ufmg.br/idp/login.jsp>

8 Código

O código deve ser obrigatoriamente escrito na **linguagem C** ou **C++**. A utilização da biblioteca STL do C++ é permitida apenas para estruturas vistas em disciplinas anteriores (AEDS1 e AEDS2). Os algoritmos ensinados em sala de aula devem ser implementados sem o auxílio da biblioteca STL. Ele deve compilar e executar corretamente nas máquinas Linux² dos laboratórios de graduação.

- O utilitário **Make** deve ser utilizado para auxiliar a compilação, um arquivo *Makefile* deve portanto ser incluído no código submetido.

Utilize as *flags*: **-Wall -Wextra -Werror -std=c99** para executar os seus experimentos em C. Caso contrário não utilize tais *flags*

O utilitário deverá gerar um executável com o nome **tp3** que deverá **obrigatoriamente** ser capaz de receber o nome do arquivo de entrada e do arquivo de saída. O comando para executá-lo deverá seguir o exemplo abaixo:

```
./tp3 input.txt output.txt LIM_M
```

 (4)

- As estruturas de dados devem ser **alocadas dinamicamente** e o código deve ser **modularizado** (divisão em múltiplos arquivos fonte e uso de arquivos cabeçalho *.h*)
- **Variáveis globais** devem ser evitadas.
- Parte da correção poderá ser feita de forma automatizada, portanto **siga rigorosamente os padrões de saída especificados**, caso contrário sua nota pode ser prejudicada.
- **Legibilidade e boas práticas** de programação serão avaliadas.
- **Não é permitido o uso de bibliotecas de terceiros**³.
- **Não é permitido o compartilhamento de código entre os estudantes**. Indícios de plágio serão investigados e, caso confirmados, serão severamente punidos.
- **Seja honesto**, você não aprende nada copiando código de terceiros nem pedindo a outra pessoa que faça o trabalho por você.

Bom trabalho!

²http://www.crc.dcc.ufmg.br/infraestrutura/laboratorios/labs_unix

³Dúvidas quanto a utilização de uma biblioteca em específico deverá ser sanada com o monitor responsável.

Referências

- [1] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.
- [2] Steven Halim and Felix Halim. *Competitive Programming 3*. Lulu Independent Publish, 2013.
- [3] Nivio Ziviani. Projeto de algoritmos com implementação em pascal e c. 2^a. *São Paulo: Thomson*, 2004.