



Rec. da 1ª Avaliação Individual

Estruturas de Dados I

Prof. Leandro M. Zatesko

19 de março de 2015

- Este instrumento avaliativo tem início às 19:20 e fim às 20:50. Durante e apenas durante este tempo, o estudante deve resolver as questões, colocando suas soluções num único arquivo de nome `NomeCompletoDoEstudante.zip` na Área de Trabalho, sendo `NomeCompletoDoEstudante` o nome completo do estudante sem diacríticos nem espaços (e.g. o arquivo de Cica Guimarães deve se chamar `CicaGuimaraes.zip`).
- Esta folha de questões possui 3 páginas (frente e verso).
- Nenhum material pode ser consultado, embora todos os programas e aplicativos instalados no ambiente que não façam uso da Internet possam ser usados. Apenas as máquinas do Laboratório podem ser usadas.
- Caso o estudante deseje fazer uso das folhas de rascunho, deve solicitar ao professor.
- Sobre a mesa são permitidos apenas lápis, canetas e borrachas. Quaisquer outros objetos, como estojos, capacetes, bolsas e mochilas, devem ser acomodados no chão ou na prateleira sobre a mesa.
- O estudante que precisar ir ao banheiro poderá fazê-lo apenas solicitando ao professor. Contudo, não será permitido que mais de um estudante esteja ausente do Laboratório ao mesmo tempo.
- Após terminar sua avaliação, o estudante deverá, permanecendo sentado, solicitar ao professor a submissão de seu arquivo `.zip`.
- Para operações de leitura e impressão, seus códigos devem sempre considerar os dispositivos padrões de entrada e saída (`stdin` e `stdout`, respectivamente).
- O estudante que tentar de algum modo *hackear* o bloqueio da Internet, reiniciar sua máquina ou sua sessão ou fraudar o instrumento avaliativo de algum modo, ou que descumprir alguma das regras estabelecidas neste cabeçalho, terá sua nota imediatamente anulada.

QUESTÃO 1 (2,5 pontos). Considere o seguinte algoritmo, que, recebendo duas matrizes quadradas A e B de ordem n , devolve a matriz $C = A + B$. No pseudocódigo, x_{ij} ($1 \leq i, j \leq n$) denota o valor da célula de linha i e coluna j na matriz X .

```
SOMEATRIZES(A,B):  
1  para i de 1 até n, faça:  
2    para j de 1 até n, faça:  
3       $c_{ij} \leftarrow a_{ij} + b_{ij}$ ;  
4  devolva C.
```

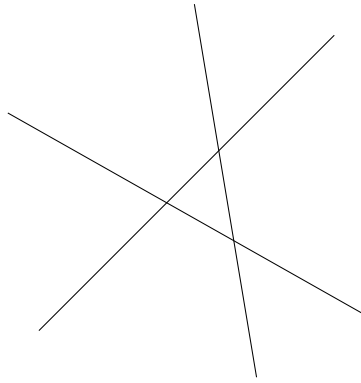
Algoritmo 1

Apresente uma equação da forma

$$T(_) = O(_)$$

que expresse a complexidade de tempo do algoritmo em função do tamanho da entrada. Depois, classifique o algoritmo como linear, quadrático, logarítmico ou exponencial. Não é necessário justificar. Apresente sua resposta num arquivo nomeado `questao1.pdf` no seu `.zip`.

QUESTÃO 2 (2,5 pontos). O maior número de regiões que é possível definir num plano com 0 retas é 1: o próprio plano — lembre-se que planos e retas são objetos geométricos ilimitados. O maior número de regiões que é possível definir num plano com 1 reta é 2. O maior número de regiões que é possível definir num plano com 2 retas é 4. O maior número de regiões que é possível definir num plano com 3 retas é 7, como ilustra a figura.



Sobre este problema, reflita (não é necessário apresentar as respostas para estas reflexões):

1. Suponha que você já possui 4 retas no plano maximizando o número de regiões definidas e que você está para colocar uma 5ª reta. Qual o maior número de retas que já estão no plano que é possível a 5ª reta interceptar? Qual o número de regiões novas que são criadas com isso?
2. Suponha que você já possui 5 retas no plano maximizando o número de regiões definidas e que você está para colocar uma 6ª reta. Qual o maior número de retas que já estão no plano que é possível a 6ª reta interceptar? Qual o número de regiões novas que são criadas com isso?
3. Suponha que você já possui $n-1$ retas no plano maximizando o número de regiões definidas e que você está para colocar uma n -ésima reta. Qual o maior número de retas que já estão no plano que é possível a n -ésima reta interceptar? Qual o número de regiões novas que são criadas com isso?
4. Sendo $R(n)$ o maior número de regiões que é possível definir com n retas no plano, escreva uma equação recorrente para $R(n)$. Dica: a base da recorrência é $R(0) = 1$, e você deve, com base nas reflexões anteriores, escrever $R(n)$ em função de $R(n-1)$.

Escreva uma função *recursiva* em C de protótipo

```
int R(int n);
```

que, recebendo um inteiro n ($0 \leq n \leq 10^9$), devolva $R(n)$. Apresente seu código num arquivo nomeado `questao2.c` (não `questao2.cpp`) no seu `.zip`, contendo exclusivamente a função `R()`.

QUESTÃO 3 (2,5 pontos). Sendo p um número primo e $a \in \{1, \dots, p-1\}$, dizemos que um inteiro $b \in \{1, \dots, p-1\}$ é o *inverso multiplicativo* de a módulo p se $a \cdot b \bmod p = 1$. Todos os números em $\{1, \dots, p-1\}$ possuem inverso multiplicativo. Por exemplo, módulo $p = 5$, 1 é o inverso multiplicativo de 1, 2 é o inverso multiplicativo de 3, 3 é o inverso multiplicativo de 2 e 4 é o inverso multiplicativo de 4. Um resultado famoso da Matemática, conhecido como *Pequeno Teorema de Fermat*, nos traz que, para encontrarmos o inverso multiplicativo de um inteiro $a \in \{1, \dots, p-1\}$, basta elevarmos a à $(p-2)$ -ésima potência, tirando o resto da divisão por p a cada multiplicação. Noutras palavras,

$$a \cdot a^{p-2} \bmod p = a^{p-1} \bmod p = 1$$

Escreva uma função em C de protótipo

```
long long invmult(long long a, long long p);
```

que, recebendo dois inteiros a e p ($1 \leq a, p < 2^{31}$), sendo p um primo e $a \in \{1, \dots, p-1\}$, devolva o inverso multiplicativo de a módulo p . Sua solução deve usar Exponenciação Binária e sua função pode chamar outras funções, desde que todas elas estejam devidamente implementadas no seu código. Apresente seu código num arquivo nomeado `questao3.c` (não `questao3.cpp`) no seu `.zip`. Seu código deve conter exclusivamente a função `invmult()` e todas as funções que são chamadas pela função `invmult()`.

QUESTÃO 4 (2,5 pontos). Escreva um programa em C que lê da entrada padrão um inteiro n ($1 \leq n \leq 10^6$) seguido de n linhas, contendo cada uma uma cadeia de caracteres, e imprime as mesmas linhas lidas, mas trocando as vogais:

- a ou A por 4;
- e ou E por 3;
- i ou I por 1;
- o ou 0 por 0 (zero);
- u ou U por v (sempre minúsculo).

Por exemplo, ao ler

3
Leandro
Miranda
Zatesko

seu programa deverá imprimir

L34ndr0
M1r4nd4
Z4t3sk0

Apresente seu código num arquivo nomeado `questao4.c` (não `questao4.cpp`) no seu `.zip`.