



# 2ª 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 2 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). Escreva um programa em C puro que:

- leia uma linha contendo dois números inteiros  $N$  e  $Q$  ( $1 \leq N \leq 10^5$ ,  $1 \leq Q \leq 10^5$ );
- em seguida, leia outra linha contendo  $N$  números inteiros  $x_i$  ( $-10^9 \leq x_i \leq 10^9$ ,  $0 \leq i < N$ ), não necessariamente ordenados, armazenando-os num vetor e ordenando o vetor em ordem não-decrescente logo em seguida;
- por último, lê  $Q$  consultas, uma por linha, cada um consistindo de um único inteiro  $\alpha$  a ser buscado no vetor, e para cada consulta imprime uma linha contendo o vetor ordenado precedendo alguma (não importa qual) ocorrência da chave  $\alpha$  buscada (se existe) por um símbolo de  $*$  — não é necessário ler todas as consultas primeiro para depois imprimir as respostas, seu programa pode imprimir a resposta para uma consulta assim que a lê.

Seu programa deverá ordenar o vetor com a função `qsort()` da `stdlib.h` e atender às consultas com busca binária, para a qual você poderá usar a função `bsearch()` da `stdlib.h` ou fornecer uma implementação própria.

Exemplo de entrada	Uma saída possível	Outra saída possível
10 3	-8 0 0 1 2 5 *15 17 23 43	-8 0 0 1 2 5 *15 17 23 43
-8 1 43 0 5 23 17 15 0 2	-8 *0 0 1 2 5 15 17 23 43	-8 0 *0 1 2 5 15 17 23 43
15	-8 0 0 1 2 5 15 17 23 43	-8 0 0 1 2 5 15 17 23 43
0		
3		

Apresente seu código num único arquivo nomeado `questao1.c` (não `questao1.cpp`).

**QUESTÃO 2** (2,5 pontos). Escreva uma função em C de protótipo

```
unsigned long long chao_de_log(unsigned long long n, unsigned long long b);
```

que recebe dois inteiros  $n$  e  $b$  ( $1 \leq n < 2^{64}$ ,  $1 < b < 2^{32}$ ) e calcula  $\lceil \log_b n \rceil$  através de uma busca binária, devolvendo o resultado. Tome cuidado ao definir o intervalo inicial da busca, ou você terá problemas com *overflow*. Para definir o intervalo inicial da busca sem afetar a complexidade da busca binária, use as ideias dos Exercícios 1 e 3 da Lista 4. Para calcular potências, utilize exponenciação binária. Sua solução deverá ser apresentada num único arquivo nomeado `questao2.c` (não `questao2.cpp`), contendo exclusivamente as implementações de `chao_de_log()` e da sua função de exponenciação binária.

**QUESTÃO 3** (2,5 pontos). Primeiro reflita:

1. Quantas vezes é possível dividir 10 por 3, descartando o resto, até ficar com 0? Quanto é  $\lceil \log_3 10 \rceil + 1$  (utilize uma calculadora do seu ambiente)? Como seria 10 escrito na base 3? Quantas vezes é possível dividir 625 por 5, descartando o resto, até ficar com 0? Quanto é  $\lceil \log_5 625 \rceil + 1$  (utilize uma calculadora do seu ambiente)? Como seria 625 escrito na base 5?
2. Qual seria o maior número inteiro  $n$  possível de se escrever com  $m$  dígitos ( $m > 0$ ) numa base  $b$  ( $b > 1$ ) qualquer? Quantos dígitos teria um inteiro positivo  $n$  quando escrito numa base  $b$  ( $b > 1$ ) qualquer? Tendo um inteiro positivo  $n$  escrito numa base  $b$ , o que acontece quando tomamos o quociente da divisão de  $n$  por  $b$ ?
3. Quantas vezes é possível dividir  $n$  por  $b$ , descartando o resto, até ficar com 0?

Com base nas reflexões acima, apresente uma implementação alternativa, mais simples, sem usar busca binária, para a função

```
unsigned long long chao_de_log(unsigned long long n, unsigned long long b);
```

Sua solução deverá ser apresentada num único arquivo nomeado `questao3.c` (não `questao3.cpp`), contendo exclusivamente a implementação da função `chao_de_log()`. Não é necessário apresentar resposta para as reflexões.

**QUESTÃO 4** (2,5 pontos). Considere o algoritmo `TETODELOG`, implementado a seguir em pseudocódigo, que, recebendo dois inteiros  $n$  e  $b$  ( $n \geq 1$ ,  $b > 1$ ), devolve  $\lceil \log_b n \rceil$ .

```

EXPBIN( $a, x$ ):
1  se  $x = 0$ , devolva 1;
2  se  $x$  é ímpar, devolva  $a \cdot \text{EXPBIN}(a, x - 1)$ ;
3   $y \leftarrow \text{EXPBIN}(a, x/2)$ ;
4  devolva  $y \cdot y$ .
TETODELOG( $n, b$ ):
1   $inf \leftarrow -1$ ;  $sup \leftarrow n$ ;
2  enquanto  $sup - inf > 1$ , faça:
3     $med \leftarrow inf + (sup - inf)/2$ ;
4     $y \leftarrow \text{EXPBIN}(b, med)$ ;
5    se  $y = n$ , devolva  $med$ ;
6    se  $y < n$ , então,  $inf \leftarrow med$ ;
7    senão,  $sup \leftarrow med$ ;
8  devolva  $sup$ .
```

### Algoritmo 1

Sendo  $m$  o número de *bits* na representação binária de  $n$  e  $k$  o número de *bits* na representação binária de  $b$ , apresente a complexidade de tempo do algoritmo `TETODELOG` em função de  $m$  e de  $k$ , usando a notação  $O$ . Não é necessário justificar. Sua solução deverá ser apresentada num arquivo PDF nomeado `questao4.pdf`.