

IMD0029 - Estrutura de Dados Básicas 1 – 2017.2

Prof. Eiji Adachi M. Barbosa

Atividade Avaliativa Prática em Laboratório – Recursão & Algoritmos de Busca

ANTES DE COMEÇAR, leia atentamente as seguintes instruções:

- Esta é uma atividade de caráter individual e sem consultas a pessoas ou material (impresso ou eletrônico).
- A atividade vale 10,0 pontos, com peso de 20% na 1ª unidade, e o valor de cada questão é informado no seu enunciado.
- Celulares e outros dispositivos eletrônicos devem permanecer desligados durante toda a prova.
- Desvios éticos ou de honestidade levarão à anulação da atividade do candidato (nota igual a zero).
- Junto a este enunciado, você também recebeu uma estrutura de diretórios contendo um diretório para cada questão. Em cada um destes diretórios, já existe uma assinatura de função e uma função main com um pequeno teste executável. A solução da sua questão deverá seguir a assinatura da função já estabelecida. Ou seja, não mude esta assinatura. Se necessário, crie funções auxiliares com outras assinaturas, mas **não mude a assinatura da função original!**

Questão 1 (20%): Considere V um array de n números inteiros distintos. Se $i < j$ e $V[i] > V[j]$ então o par (i, j) é chamado uma inversão de V . Por exemplo, o array $V[4] = \{10, 30, 20, 25\}$ possui duas inversões: (I) $(1, 2)$ e (II) $(1, 3)$, ou seja, (I) $1 < 2$ & $V[1] > V[2]$, (II) $1 < 3$ & $V[1] > V[3]$. Neste contexto, implemente uma função que recebe um array de entrada V e retorna um inteiro X com o número de inversões que existem em V .

Questão 2 (30%): Implemente o algoritmo de ordenação MergeSort.

Questão 3 (50%): Implemente o algoritmo de ordenação QuickSort que usa o algoritmo de partição que leva em consideração a existência de elementos repetidos.

ENTREGÁVEL

O entregável desta atividade deverá seguir a mesma estrutura de diretórios do código fonte que você recebeu com este enunciado, obviamente, contendo os arquivos fonte utilizados para construir sua solução nos diretórios de cada questão. Além disso, o diretório pai deverá ter o seu nome e matrícula, seguindo o padrão `<PRIMEIRO>_<SOBRENOME>-<MATRICULA>`. Por exemplo:

```
> JOAO_SILVA-200012345
> q1
> q2
> q3
> q4
```

Toda esta estrutura de diretórios, incluindo os arquivos fonte com sua solução, deverá ser compactada num arquivo .zip que também deverá seguir o padrão `<PRIMEIRO>_<SOBRENOME>-<MATRICULA>`. Este arquivo compactado deverá ser entregue via SIGAA até as **22:15. Este é um prazo fixo que não será estendido**, exceto em casos muito excepcionais (ex.: SIGAA fora do ar). Ou seja, entregas após este horário não serão aceitas. A atividade do SIGAA permite apenas um envio, portanto certifique-se de que está enviando a versão correta antes de anexar ao SIGAA.

CRITÉRIOS DE CORREÇÃO

Para a correção desta atividade, serão levados em consideração, dentre outros, os seguintes pontos:

- Obediência às regras definidas para as assinaturas de função e para o entregável (arquivo .zip), conforme especificado no enunciado desta atividade
- Existência de erros ou warnings de compilação do código fonte¹

¹ Compile usando as flags `-Wall -pedantic -std=c++11`

- Programas executam sem apresentar falhas e produzem os resultados esperados
- Soluções atendem critérios de complexidade, caso estabelecido no enunciado
- Apresentação e organização do código fonte entregue (identação, nome das variáveis, modularização do código em função, etc)

Obs.: Para cada questão, já há uma função main com um pequeno teste executável. Este é um teste simples que **não garante** a corretude da sua implementação. Ou seja, se sua implementação passou no teste executável disponibilizado junto a este enunciado, isto é apenas uma evidência mínima de que ela está correta. Para fins de correção, eu utilizarei outra bateria de testes mais completa, além de analisar manualmente o código produzido.

ENTREGÁVEL

O entregável desta atividade deverá seguir a mesma estrutura de diretórios do código fonte que você recebeu com este enunciado, obviamente, contendo os arquivos fonte utilizados para construir sua solução nos diretórios de cada questão. Além disso, o diretório pai deverá ter o seu nome e matrícula, seguindo o padrão <PRIMEIRO>_<SOBRENOME>-<MATRICULA>. Por exemplo:

```
> JOAO_SILVA-200012345
> q1
> q2
> q3
> q4
```

Toda esta estrutura de diretórios, incluindo os arquivos fonte com sua solução, deverá ser compactada num arquivo .zip que também deverá seguir o padrão <PRIMEIRO>_<SOBRENOME>-<MATRICULA>. Este arquivo compactado deverá ser entregue via SIGAA até as **22:15**. **Este é um prazo fixo que não será estendido**, exceto em casos muito excepcionais (ex.: SIGAA fora do ar). Ou seja, entregas após este horário não serão aceitas. A atividade do SIGAA permite apenas um envio, portanto certifique-se de que está enviando a versão correta antes de anexar ao SIGAA.

CRITÉRIOS DE CORREÇÃO

Para a correção desta atividade, serão levados em consideração, dentre outros, os seguintes pontos:

- Obediência às regras definidas para as assinaturas de função e para o entregável (arquivo .zip), conforme especificado no enunciado desta atividade
- Existência de erros ou warnings de compilação do código fonte²
- Programas executam sem apresentar falhas e produzem os resultados esperados
- Soluções atendem critérios de complexidade, caso estabelecido no enunciado
- Apresentação e organização do código fonte entregue (identação, nome das variáveis, modularização do código em função, etc)

Obs.: Para cada questão, já há uma função main com um pequeno teste executável. Este é um teste simples que **não garante** a corretude da sua implementação. Ou seja, se sua implementação passou no teste executável disponibilizado junto a este enunciado, isto é apenas uma evidência mínima de que ela está correta. Para fins de correção, eu utilizarei outra bateria de testes mais completa, além de analisar manualmente o código produzido.

² Compile usando as flags `-Wall -pedantic -std=c++11`