



UNIVERSIDADE FEDERAL DE VIÇOSA

CAMPUS RIO PARANAÍBA

Instituto de Ciências Exatas e Tecnológicas

Campus Universitário – Cx. Postal 22 - Rio Paranaíba, MG – 38810-000

SIN 213 Projeto de Algoritmos - Prova 1 (08/03/2021) – 20 Pontos

Instruções para esta avaliação:

- **Sobre a duração desta avaliação:**
 - O prazo máximo de entrega desta prova é dia **14/03/2021** as **23:59h** pelo PVANet.
- **Sobre o desenvolvimento dos códigos:**
 - **Todos** os algoritmos devem ser implementados na linguagem **Python** sem bibliotecas externas (Que necessitem de importação).
 - A entrega dos códigos desenvolvidos será feita pelo PVANet em um arquivo compactado com o formato **"Matrícula.zip"**, contendo os códigos desenvolvidos e um arquivo texto puro nomeado **Links.txt**, com o link para os vídeos do youtube contendo cada questão descrita abaixo.
 - **Todos** os algoritmos implementados farão uso de **valores inteiros** como entrada, ou seja, vetores de inteiros, variáveis com inteiros, etc.
 - Utilize os valores do vetor no arquivo **template.py** (<https://clausiusreis.github.io/> em **PROVA 1**) para carregar todas as estruturas de dados desta avaliação.
 - Será verificado plágio entre os exercícios, particularmente sobre as funções solicitadas.
- **Sobre a gravação dos vídeos:**
 - Cada vídeo poderá ter duração **máxima** de **15 minutos**.
 - O rosto de cada aluno deve ser visível durante todo o vídeo.
 - Não será aceito o envio dos arquivos dos vídeos, apenas o link (Youtube).
- **Método avaliativo:**
 - A qualidade da sua câmera não será avaliada, mas o material apresentado deve estar legível e o rosto do aluno apresentando deve aparecer durante toda a explicação.
 - Será avaliado: (10 pontos para cada questão)
 - **No vídeo:**
 - Se o aluno demonstrou domínio do conteúdo ou somente leu um texto. (2 pontos)
 - Qualidade da explicação e correteza das informações apresentadas (Ex. código e complexidade). O aluno deve demonstrar uma linha de raciocínio. (3 pontos)
 - Capacidade de síntese. O aluno deve ser capaz de explicar o conteúdo com clareza e no prazo máximo de 15 minutos. (1 ponto)
 - Qualidade do material apresentado. (1 ponto)
 - **No código:**
 - Código implementado, incluindo clareza e organização do mesmo. (3 pontos)

Questão 1:

- Implemente uma estrutura de dados do tipo **lista circular duplamente encadeada**.
- Carregue o vetor do arquivo **template.py** para dentro das estruturas de dados.
- Crie uma função para pesquisar uma chave X do tipo inteiro nesta lista, partindo dos nós 0 (zero) e N/2 simultaneamente. As buscas iniciando de cada nó devem ocorrer simultaneamente nos dois sentidos da lista.
- A busca termina quando a chave X for encontrada ou toda a lista for percorrida, retornando uma busca negativa.
- Grave um vídeo explicando a implementação da função de busca, e apresente a função custo $C(n)$ dessa função para o pior caso, melhor caso e caso médio. Considere as comparações para o cálculo de complexidade. Qual o comportamento assintótico $O(C(n))$ da função desenvolvida?
- Coloque este vídeo gravado no seu canal do Youtube em modo “**não listado**” e coloque o link no arquivo **Link.txt** no mesmo diretório do código da questão 1.

Questão 2:

- Implemente os algoritmos de Heapsort e Árvore binária de pesquisa (**Não balanceada**).
- Carregue o vetor do arquivo **template.py** para dentro das estruturas de dados.
- Implemente uma função de busca para cada algoritmo que percorra a estrutura de dados a procura de uma chave X.
- Grave um vídeo explicando a implementação da função de busca de ambos os algoritmos, e apresente o custo $C(n)$ para a função busca no algoritmo de Heapsort (considerando as comparações) para o pior caso, melhor caso e caso médio. Uma vez que o Heapsort simula o comportamento de uma árvore binária, explique qual das duas funções de busca é mais eficiente e o motivo dessa eficiência.
- Coloque este vídeo gravado no seu canal do Youtube em modo “**não listado**” e coloque o link no arquivo **Link.txt** no mesmo diretório do código da questão 2.