

SUBSTITUIÇÃO DAS ATIVIDADES ATRASADAS (COM DESCONTO DE NOTA POR CONTA DO ATRASO)

Atividade 1

Essa atividade visa dar os primeiros passos em implementação python pelo colab do conteúdo de listas.

Peço que implementem um novo notebook no google colab com as classes No e Lista que foram vistas em aula e com novos métodos:

Método que imprime apenas os três primeiros elementos da lista;

Método que insere um elemento no início da lista;

Método que remove o primeiro elemento da lista;

Método que remove os números pares da lista;

Método que retorna o tamanho da lista.

O código do notebook do google colab deve ser compartilhado por meio de link em que qualquer pessoa da UnDF possa visualizar. A entrega da sua atividade é o link para o seu código.

Atividade 2

Essa atividade visa conectarmos os conteúdos vistos nas duas primeiras semanas de aula.

Peço que redefinam a classe pilha começando pelo seu único atributo data, em vez de usar um vetor do python, usar uma instância de Lista. Para isso, primeiramente deverão ser implementados novos métodos na classe Lista, tais como:

1. Imprimir o último elemento da lista
2. Remover o último elemento da lista

Fazer todas as adaptações necessárias na classe Pilha (seu atributo e métodos).

Lembra de usar todos os métodos possíveis que já tinham sido implementados na classe Lista na semana passada durante a aula e da tarefa anterior também. Por gentileza usar um notebook no google colab exclusivo para essa atividade, não esqueça de identificar o seu código com seu nome, e escrever objetivos, assim como fazer testes.

O código do notebook do google colab deve ser compartilhado por meio de link em que qualquer pessoa da UnDF possa visualizar. A entrega da sua atividade é o link para o seu código.

Atividade 3

Essa atividade visa conectarmos os conteúdos vistos até esse momento.

Peço que redefinam a classe fila começando pelo seu único atributo data, em vez de usar um vetor do python, usar uma instância de Lista. Para isso, devem ser implementados novos métodos na classe Lista, descubra quais são.

Fazer todas as adaptações necessárias na classeFila (seu atributo e métodos).

Quando estiver com a classe Fila redefinida, implementar uma solução para o seguinte problema:

- Dado um arquivo de texto de nome "inscritos.txt" contendo uma lista de nomes (string) e a categoria de atendimento (normal: 'n', ou prioritário: 'p'), com o separador vírgula em cada linha (Ex: Fulano de Tal, 'p'); implemente um código em python que leia esse arquivo e gere uma lista de atendimentos a serem realizados em um determinado órgão público.
- A ordem de atendimento deve ser: a cada cinco prioritários que saem da fila, três normais também são atendidos e assim sucessivamente. Ou seja, seguindo a lógica p, p, p, p, p, n, n, n, p, p, p, p, p, n, n, n...
- Dentro de cada categoria a ordem é a mesma lida no arquivo.
- Os atendimentos são realizados de meia em meia hora, começando o primeiro às 11h00 e terminando o último às 15h30, dentro de um mesmo dia, ou seja, o horário de atendimento deve ser impresso, junto com o nome e a categoria.
- Caso a lista tenha mais nomes que a quantidade de atendimentos possíveis dentro de um dia de expediente, a solução deve imprimir o atendimento para quantos dias que forem necessários, colocando um texto "1o dia", "2o dia", ... e assim por diante, antes de listar os atendimentos de cada dia.
- O resultado do código deve ser salvo em um arquivo de nome "ordemAtendimento.txt". Esse arquivo deve ter a lista com a ordem de atendimento com a informação do dia, do horário, do nome da pessoa e da sua categoria, em cada linha.

Estou compartilhando um arquivo de exemplo de input para o código.

Por gentileza usar um notebook no google colab exclusivo para essa atividade, não esqueça de identificar o seu código com seu nome, e escrever objetivos, assim como fazer testes.

O código do notebook do google colab deve ser compartilhado por meio de link em que qualquer pessoa da UnDF possa visualizar. A entrega da sua atividade é o link para o seu código.

Atividade 4

Essa atividade visa estudarmos um pouco mais os conteúdos vistos na última aula, assim como prepará-los para o que será visto nas próximas.

Para os cinco algoritmos de ordenação vistos em aula:

Faça a versão de ordenação em ordem decrescente de cada um.

Para um vetor de tamanho dez, formado por números aleatórios (desordenados), calcule o tempo de execução de cada algoritmo (use uma função pronta do python para medir esse tempo).

Repita a medida dos tempos do item anterior, mas agora com um vetor de tamanho vinte.

Compare os tempos, tire conclusões sobre os resultados encontrados e coloque em células de texto do colab.

Por gentileza usar um notebook no google colab exclusivo para essa atividade, não esqueça de identificar o seu código com seu nome, e escrever objetivos, assim como fazer testes.

O código do notebook do google colab deve ser compartilhado por meio de link em que qualquer pessoa da UnDF possa visualizar. A entrega da sua atividade é o link para o seu código.

Essa atividade poderá ser feita em duplas ou individualmente, a critério de cada um. Caso o trabalho seja em dupla, apenas um precisa enviar o link (mas não esqueça de colocar os nomes de quem fez o trabalho no notebook colab).

Atividade 5

Essa atividade visa estudarmos um pouco mais os conteúdos vistos em aula, a ideia é implementar uma Tabela de Dispersão em Python.

Para fazer isso, será necessário um estudo sobre estrutura de dicionário, ou estrutura chave-valor em python. Pesquisem sobre isso antes de iniciar a implementação.

A implementação da tabela hash, consistirá em criação de uma classe, com dois atributos, o tamanho da tabela (que deverá ser passado como argumento da classe) e o dicionário vazio. Os métodos a serem implementados são o de inserção de novos elementos na tabela, fazendo o tratamento de colisões (por pelos três métodos de endereçamento aberto); o de busca por um elemento; e o de remoção de um elemento. Métodos adicionais como a impressão de certas linhas da tabela ou outra funcionalidade que seja útil são bem-vindos.

Faça testes, comente o código, use os textos entre os códigos para explicar o que tem em cada chunk, tire conclusões sobre os resultados encontrados e coloque em células de texto do colab.

Por gentileza usar um notebook no google colab exclusivo para essa atividade, não esqueça de identificar o seu código com seu nome, e escrever objetivos, assim como fazer testes.

O código do notebook do google colab deve ser compartilhado por meio de link em que qualquer pessoa da UnDF possa visualizar. A entrega da sua atividade é o link para o seu código.

Essa atividade poderá ser feita em duplas ou individualmente, a critério de cada um. Caso o trabalho seja em dupla, apenas um integrante precisa enviar o link (mas não esqueça de colocar os nomes de quem fez o trabalho no notebook colab).

Atividade 6

Essa atividade visa estudarmos um pouco mais os conteúdos sobre árvores vistos em aula e no material. A ideia é implementar uma estrutura de dados em árvore em Python.

Peço que implementem um novo notebook no google colab com as seguintes classes:

- Nó (com os atributos chave, filho esquerdo e filho direito), e
- Árvore (com o atributo raiz (que tem o tipo nó))

A árvore deve ter os seguintes métodos:

- Imprimir o elemento de máxima prioridade em uma árvore min-heap
- Imprimir os elementos de uma árvore min-heap em um vetor
- Inserir nó em uma árvore min-heap
- Extrair o mínimo em uma árvore min-heap
- Inserir nó em uma árvore binária de busca
- Remover um nó em uma árvore binária de busca
- Imprimir os elementos de uma árvore binária de busca em um vetor segundo cada uma das estratégias pré-ordem, em ordem e pós-ordem.
- Verificar se a árvore é min-heap
- Verificar se a árvore é binária de busca

Faça testes, comente o código, use os textos entre os códigos para explicar o que tem em cada chunk.

Por gentileza usar um notebook no google colab exclusivo para essa atividade, não esqueça de identificar o seu código com seu nome, e escrever objetivos, assim como fazer testes.

O código do notebook do google colab deve ser compartilhado por meio de link em que qualquer pessoa da UnDF possa visualizar. A entrega da sua atividade é o link para o seu código.

Essa atividade poderá ser feita em duplas ou individualmente, a critério de cada um. Caso o trabalho seja em dupla, apenas um integrante precisa enviar o link (mas não esqueça de colocar os nomes de quem fez o trabalho no notebook colab).

Atividade 7

Essa atividade visa estudarmos um pouco mais os conteúdos sobre árvores vistos em aula e no material. Novamente, a ideia é implementar uma estrutura de dados em árvore em Python.

Peço que implementem um novo notebook no google colab com as seguintes classes:

- Nó (com os atributos chave, altura, fator de balanceamento, pai, filho esquerdo e filho direito), e
- Árvore (com o atributo raiz (que tem o tipo nó))

A árvore deve ter os seguintes métodos:

- Inserir nó em uma árvore AVL
- Remover um nó em uma árvore AVL

Repare que para programar os métodos de inserção e remoção em árvore AVL, são necessários outros métodos adicionais que mantenham a árvore AVL balanceada e que também devem ser incorporados à classe árvore. Faça testes, comente o código, use os textos entre os códigos para explicar o que tem em cada chunk.

Por gentileza usar um notebook no google colab exclusivo para essa atividade, não esqueça de identificar o seu código com seu nome, e escrever objetivos, assim como fazer testes.

O código do notebook do google colab deve ser compartilhado por meio de link em que qualquer pessoa da UnDF possa visualizar. A entrega da sua atividade é o link para o seu código.

Essa atividade poderá ser feita em duplas ou individualmente, a critério de cada um. Caso o trabalho seja em dupla, apenas um integrante precisa enviar o link (mas não esqueça de colocar os nomes de quem fez o trabalho no notebook colab).

Atividade 8

Essa atividade visa estudarmos os conteúdos sobre grafos vistos em aula e no material. A ideia é implementar uma estrutura de dados em grafo em Python. Peço que implementem um novo notebook no google colab com a seguinte classe:

- Grafo (com o atributo dic (que tem o tipo dicionário)) (o dicionário inicializa vazio, mas a ideia é mapear a cada nó a sua lista de vizinhos).

O grafo deve ter os seguintes métodos:

- Inserir aresta com um nó origem e um destino no grafo
- Calcula o caminho mais curto entre dois nós (usar busca em largura)

Faça testes, comente o código, use os textos entre os códigos para explicar o que tem em cada chunk.

Por gentileza usar um notebook no google colab exclusivo para essa atividade, não esqueça de identificar o seu código com seu nome, e escrever objetivos, assim como fazer testes.

O código do notebook do google colab deve ser compartilhado por meio de link em que qualquer pessoa da UnDF possa visualizar. A entrega da sua atividade é o link para o seu código.

Essa atividade poderá ser feita em duplas ou individualmente, a critério de cada um. Caso o trabalho seja em dupla, apenas um integrante precisa enviar o link (mas não esqueça de colocar os nomes de quem fez o trabalho no notebook colab).

Atenciosamente

Fernando Madeira