

Data	Tema	Objetivo Pedagógico
03/03/16	Pilhas -	<ul style="list-style-type: none"> · Utilizar structs, ponteiros e alocação dinâmica para implementar ou corrigir um programa que utilize as funções de <i>inserção</i> e <i>retirada</i> de elementos em uma PILHA de tamanho fixo. · Identificar cenários em que o uso de uma PILHA é recomendado. · Estimar a complexidade computacional dos algoritmos de inserção e retirada de uma PILHA.
10/03/16	Fila	<ul style="list-style-type: none"> · Utilizar structs, ponteiros e alocação dinâmica para implementar ou corrigir um programa que utilize as funções de <i>inserção</i>, e <i>retirada</i> de elementos em uma FILA de tamanho fixo. · Identificar cenários em que o uso de uma FILA é recomendado. · Estimar a complexidade computacional dos algoritmos de inserção e retirada de UMA FILA.
17/03/16	Lista Simples	<ul style="list-style-type: none"> · utilizar ponteiros, structs e alocação dinâmica para implementar ou corrigir um programa que utilize as funções de <i>inserção</i>, <i>busca</i> e <i>retirada</i> de elementos em uma LISTA SIMPLES de tamanho variável. · Identificar cenários em que o uso de uma LISTA SIMPLES é recomendado. · Estimar a complexidade computacional dos algoritmos de inserção, busca e retirada de uma LISTA SIMPLES.
24/03/16	Sem aula	· Semana sem aula
31/03/16	Lista Dupla	<ul style="list-style-type: none"> · utilizar ponteiros, structs e alocação dinâmica para implementar ou corrigir um programa que utilize as funções de <i>inserção</i>, <i>busca (nos dois sentidos)</i> e <i>retirada</i> de elementos em uma LISTA DUPLA de tamanho variável. · Identificar cenários em que o uso de uma LISTA DUPLA é recomendado. · Estimar a complexidade dos algoritmos de inserção, busca e retirada de uma LISTA DUPLA. · Utilizar ponteiros, structs e alocação dinâmica para implementar ou corrigir um programa que utilize as funções de <i>inserção</i>, <i>busca (nos dois sentidos)</i> e <i>retirada</i> de elementos em uma LISTA CIRCULAR de tamanho variável. · Identificar cenários em que o uso de uma LISTA CIRCULAR é recomendado. · Estimar a complexidade computacional dos algoritmos de inserção, busca e retirada de uma LISTA CIRCULAR.
07/04/16		· Revisão para a Primeira Avaliação
14/04/16		· Primeira Avaliação
28/04/16	Recursão	<ul style="list-style-type: none"> · Implementar ou corrigir um programa que utilize o conceito de recursão. · Identificar cenários em que o uso de RECURSÃO é recomendado. · Aplicar o conceito de recursão para implementar uma função que inverte a ordem de elementos em uma lista encadeada. · Aplicar o conceito de recursão para implementar uma função que reverte a ordem dos elementos de uma pilha. · Estimar a complexidade computacional de uma função recursiva.
05/05/16	Ordenação	<ul style="list-style-type: none"> · Conhecer os princípios de funcionamento de um conjunto de algoritmos de ordenação dado. · Determinar a complexidade computacional de cada um dos algoritmos do conjunto. · Aplicar o conceito de recursão para implementar ou corrigir um algoritmo de ordenação
12/05/16	Árvores Binárias	· Conhecer o conceito de ÁRVORES DE DECISÃO e sua justificativa

		para o seu uso no armazenamento de dados. · Conhecer o uso de ÁRVORES DE DECISÃO em estruturas hierárquicas. · Implementar uma ÁRVORE DE DECISÃO empregando vetores ou alocação dinâmica de memória e o conceito de recursividade. · Calcular a complexidade computacional de uma ÁRVORE DE DECISÃO profundidade determinada.
19/05/16	Balanceamento de Árvores Binárias	· Aplicar o conceito de balanceamento para equilibrar uma ÁRVORE DE DECISÃO. · Realizar buscas em uma ÁRVORE DE DECISÃO e determinar a complexidade deste algoritmo.
02/06/16	Grafos	· Implementar um problema de armazenamento de dados sob forma de GRAFO · Realizar pesquisa em um GRAFO implementado. · Calcular a menor distância entre nós de um grafo · Modelar um problema com grafos
09/06/16	Tabela Hash	· Explicar o funcionamento dos mecanismo <i>inserção, retirada, e mudança de tamanho</i> de TABELAS HASH, · Determinar a complexidade computacional de uma TABELA HASH, · Explicar o funcionamento do tratamento de colisões em uma TABELA HASH · Implementar uma tabela HASH para armazenar palavras
16/06/16		· Revisão para a 2ª Avaliação
23/06/16		· Apresentação dos Projetos
30/06/16		· Atividade de Recuperação
07/07/16		· APS - compressão de dados