



CENTRO UNIVERSITÁRIO FAMETRO

GRADUAÇÃO ANALISE DESENVOLVIMENTO DE SISTEMAS

NICOLE MAIARA ABREU DE AGUIAR 2411118

DENILSON EMANOEL COLARES BEZERRA 2400651

MIGUEL ANTONY SANTOS DE LIMA 2407566

ATIVIDADE PARCIAL

Manaus - AM

27/10/2025

1. O que significa alocação estática de memória para um conjunto de elementos?

A alocação estática de memória é aquela feita em tempo de compilação, ou seja, o tamanho da memória é definido antes da execução do programa. Em C, isso acontece com variáveis globais, locais e vetores cujo tamanho é fixo. Exemplo: `int vetor[10];` — o vetor terá sempre 10 posições.

2. Qual a diferença entre alocação estática e alocação dinâmica?

Característica	Alocação Estática	Alocação Dinâmica	Quando é definida	Compilação
Execução	Flexível	Fixa	Ajustável	

Uso típico: Vetores estáticos, Listas, filas, árvores

Funções relacionadas: `malloc`, `calloc`, `free`

3. O que é um ponteiro?

Um ponteiro é uma variável que armazena o endereço de memória de outra variável. Ele é fundamental para manipulação dinâmica de memória e estruturas encadeadas.

4. O que é uma estrutura de dados homogênea?

São estruturas em que todos os elementos têm o mesmo tipo de dado, como vetores e matrizes.

5. O que é uma estrutura de dados heterogênea?

São estruturas com diferentes tipos de dados agrupados, como `struct` em C.

6. Qual a vantagem das listas sobre vetores em termos de consumo de memória?
explique

Listas encadeadas utilizam apenas a memória necessária para armazenar os elementos existentes, enquanto vetores reservam um tamanho fixo, podendo desperdiçar memória. Exemplo: `int vetor[1000];` → ocupa memória fixa para 1000 elementos

Lista encadeada → cresce conforme insere elementos

7. O que é uma lista simplesmente encadeada? Apresente um diagrama para ilustrar essa estrutura de dados.

É uma estrutura dinâmica onde cada nó armazena um dado e um ponteiro para o próximo nó. $[10 \mid *] \rightarrow [25 \mid *] \rightarrow [30 \mid \text{NULL}]$

8. O que é uma lista duplamente encadeada? Apresente um diagrama para ilustrar essa estrutura de dados

É uma lista encadeada onde cada nó possui ponteiro para o próximo e para o anterior:

$\text{NULL} \leftarrow [* \mid \text{dado} \mid *] \leftrightarrow [* \mid \text{dado} \mid *] \leftrightarrow [* \mid \text{dado} \mid *] \rightarrow \text{NULL}$

9. O que é uma lista duplamente encadeada? Apresente um diagrama para ilustrar essa estrutura de dados

É uma variação da lista encadeada onde o último nó aponta para o primeiro, formando um ciclo $[\text{dado} \mid *] \rightarrow [\text{dado} \mid *] \rightarrow [\text{dado} \mid *]$

10. Explique o funcionamento do algoritmo de busca binária e sequencial.

Característica Busca Sequencial Busca Binária

Estratégia Varre elemento por elemento Divide o vetor ao meio

Requisito Nenhum Vetor ordenado Complexidade $O(n)$ $O(\log n)$

11. Explique o funcionamento dos seguintes algoritmos de ordenação: Insertion sort, Selection sort, Merge sort, Count sort, Quicksort

O Insertion Sort é um algoritmo de ordenação que funciona inserindo cada elemento em sua posição correta em uma lista ordenada. Ele começa com o segundo elemento da lista e compara-o com os elementos anteriores, deslocando-os para a direita até encontrar a posição correta. Em seguida, insere o elemento na

posição correta e repete o processo para todos os elementos da lista. A complexidade do Insertion Sort é $O(n^2)$ no pior caso.

O Selection Sort é um algoritmo de ordenação que funciona selecionando o menor elemento da lista e trocando-o com o primeiro elemento. Ele repete esse processo para a sublista restante, excluindo o primeiro elemento, e continua até que a lista esteja ordenada. A complexidade do Selection Sort também é $O(n^2)$ no pior caso.

O Merge Sort é um algoritmo de ordenação que funciona dividindo a lista em sublistas menores e ordenando cada sublista recursivamente. Ele divide a lista em duas sublistas de tamanho aproximadamente igual, ordena cada sublista e, em seguida, mescla as sublistas ordenadas em uma lista única ordenada. A complexidade do Merge Sort é $O(n \log n)$ no pior caso.

O Count Sort é um algoritmo de ordenação que funciona contando o número de ocorrências de cada elemento na lista. Ele cria um array de contagem com tamanho igual ao maior elemento + 1 e conta o número de ocorrências de cada elemento. Em seguida, reconstrói a lista ordenada com base no array de contagem. A complexidade do Count Sort é $O(n + k)$ no pior caso, onde k é o maior elemento.

O Quicksort é um algoritmo de ordenação que funciona selecionando um pivô e particionando a lista em duas sublistas: uma com elementos menores que o pivô e outra com elementos maiores. Ele ordena cada sublista recursivamente e combina as sublistas ordenadas e o pivô para obter a lista ordenada. A complexidade do Quicksort é $O(n \log n)$ no caso médio, mas pode ser $O(n^2)$ no pior caso.

REFERÊNCIAS:

TANENBAUM, A.; BOS, H. Sistemas Operacionais Modernos. 4ª ed. Pearson, 2015.

DEITEL, H. & DEITEL, P. C: Como Programar. 6ª ed. Pearson, 2011.

KERNIGHAN, B.; RITCHIE, D. Linguagem C. 2ª ed. Pearson, 2013.

ZIVIANI, N. Projeto de Algoritmos. 3ª ed. Cengage, 2011.

ASCENCIO, A.; CAMPOS, E. Fundamentos da Programação de Computadores. Pearson, 2008.

CORMEN, T. et al. Algoritmos – Teoria e Prática. 3ª ed. Elsevier, 2012.

GOODRICH, M.; TAMASSIA, R. Estruturas de Dados e Algoritmos em C++. LTC, 2011.

ZIVIANI, N. Projeto de Algoritmos. 3ª ed. Cengage, 2011.

DEITEL, H.; DEITEL, P. C: Como Programar. 6ª ed. Pearson, 2011.

SEEDGEWICK, R. Algorithms in C. Addison-Wesley, 1998.

HONORATO, Bruno de Almeida. Algoritmos de ordenação: análise e comparação. Devmedia, 31 mar. 2018

link do repositório:

<https://github.com/DenilsonEmanoelcop/atividadeN2estruturadedados.git>