Parte 1 – Teoria

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

A alocação estática de memória é um método de gerenciamento onde o espaço de memória é alocado e liberado em tempo de compilação, antes da execução do programa. O tamanho da memória é determinado previamente e permanece fixo durante toda a execução. Vantagens incluem simplicidade e eficiência de tempo de execução. A principal desvantagem é a limitação do espaço disponível, que não pode ser alterado dinamicamente. Exemplos comuns são variáveis e arrays estáticos.
https://techner.com.br/glossario/o-que-e-alocacao-estatica-de-memoria/

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

A alocação estática ocorre em tempo de compilação, geralmente no segmento Data, com tamanho fixo e para toda a vida do programa, aplicada a variáveis globais ou estáticas. Já a alocação dinâmica acontece em tempo de execução, na área Heap, e permite que o programa solicite, redimensione e libere blocos de memória conforme a necessidade. O gerenciamento estático é automático, enquanto o dinâmico é manual, exigindo que o programador use funções como malloc e free.
https://www.inf.ufpr.br/hexsel/ci067/10_aloc.html

3. **O que é um ponteiro?**

Um ponteiro é uma variável que armazena um endereço de memória, permitindo referenciar o local onde outra variável ou dado está armazenado. O operador de derreferência (*) é usado para acessar o conteúdo dessa posição de memória. Eles são essenciais para a alocação dinâmica (malloc/new), permitindo criar vetores e estruturas de dados como listas encadeadas, e suportam a aritmética de ponteiros para navegação eficiente.
https://www.inf.pucrs.br/~pinho/PRGSWB/Ponteiros/ponteiros.html

4. **O que é uma estrutura de dados homogêneos?**

Estruturas de dados homogêneas, como vetores (unidimensionais) e matrizes (multidimensionais), são formas de organização que permitem armazenar um grupo de valores sob um único identificador. Sua principal característica é a homogeneidade, o que significa que todos os valores armazenados devem ser do mesmo tipo de dado. Isso garante o armazenamento eficiente e o acesso aleatório aos elementos através de seus índices.
http://sae.unb.br/cae/conteudo/unbfga/apc/new_estruturadedadoshomogenea.html

5. **O que é uma estrutura de dados heterogêneos?**

Uma estrutura de dados heterogênea, ou registro (struct em C), é uma coleção de variáveis, possivelmente de tipos de dados diferentes, agrupadas sob um único nome ou identificador. Ela é usada para organizar informações logicamente relacionadas, como os dados de uma conta bancária. O acesso aos membros da estrutura é feito através do operador ponto (.), ou do operador seta (->) quando se utiliza um ponteiro.
https://sae.unb.br/cae/conteudo/unbfga/apc/new_estruturaheterogenea.html

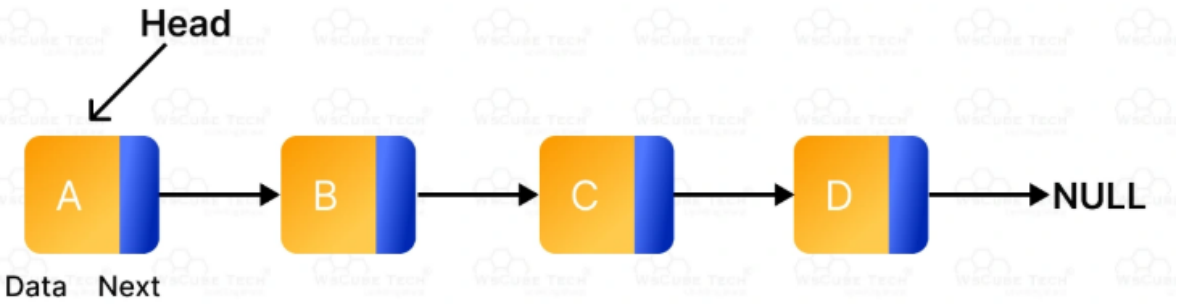
6. **Qual a vantagem das listas sobre os vetores em termos de consumo de memória? Exemplifique.**

A principal vantagem teórica das listas sobre os vetores é a complexidade de inserção e exclusão de elementos, que é O(1) para listas. Em contraste, vetores exigem reorganização contígua da memória, tornando a exclusão O(n). Por exemplo, ao deletar um

elemento, a lista só precisa atualizar um ponteiro, enquanto o vetor precisa deslocar todos os elementos seguintes.
https://horenbergerb.github.io/2024/05/09/vector-vs-list.html#random-access-and-iterating

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

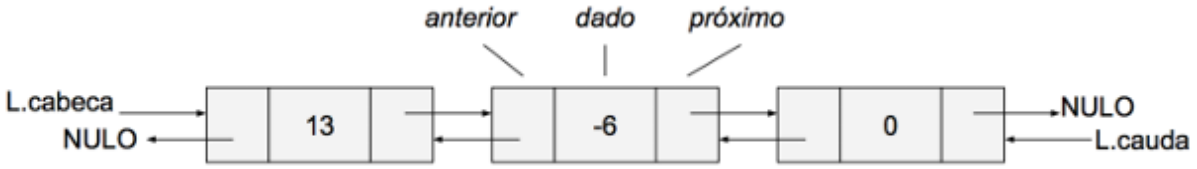
Uma lista simplesmente encadeada é uma estrutura de dados onde cada elemento, chamado nó, armazena uma porção de dados e uma referência (ponteiro) para o próximo nó na sequência. O primeiro nó é o Topo e o último aponta para Null, indicando o fim. Permite inserção e exclusão eficientes (O(1) no início) sem realocação de espaço, diferente de vetores.



https://www.wscubetech.com/resources/dsa/singly-linked-list-data-structure

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

Uma lista duplamente encadeada é uma estrutura de dados onde cada nó possui um ponteiro para o próximo nó e um ponteiro adicional para o nó anterior. Essa estrutura permite que a lista seja percorrida em ambas as direções. O nó com o ponteiro "anterior" nulo é a cabeça, e o com o "próximo" nulo é a cauda.



https://www.inf.ufpr.br/gregio/CI1001/ListaDuplamenteEncadeada.pdf

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

A busca sequencial percorre o vetor elemento por elemento, comparando-o com o valor procurado. É simples, funciona em qualquer lista, mas tem complexidade O(n). Já a busca binária exige que o vetor esteja ordenado. Ela descarta metade dos elementos a cada comparação, focando no elemento central, o que resulta em uma complexidade mais eficiente de O(log n).
https://programai.com.br/busca-binaria-e-sequencial/

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

O Insertion Sort insere elementos na sublista ordenada; o Selection Sort encontra o menor elemento restante para posicioná-lo; o Merge Sort divide, ordena e mescla as subpartes; o Quicksort particiona a lista em torno de um pivô e ordena recursivamente. Por fim, o Count Sort é um algoritmo de ordenação estável com complexidade O(n+k) (onde k é o maior valor), que determina a posição de cada elemento contando quantos elementos são menores que ele, usando um vetor auxiliar de contagem.
https://www.treinaweb.com.br/blog/conheca-os-principais-algoritmos-de-ordenacao
https://pt.wikipedia.org/wiki/Counting_sort