

Atividade Bônus Monitoria

Questão 1: O Mediano da Festa

Foco do Algoritmo: Quick Sort

História do Problema:

A organização do festival de música "Lollapalooza Jataí" precisa definir a ordem das bandas no palco principal. Cada banda tem um "índice de popularidade". Para criar um cronograma equilibrado, o organizador principal quer identificar a banda com o índice de popularidade mediano. Com a lista de popularidade de todas as N bandas, ordenar a lista inteira seria uma opção, mas com milhares de bandas, isso pode ser lento. Seu desafio é encontrar o valor mediano de forma mais eficiente.

Tarefa:

Escreva um programa em C que leia uma lista de N inteiros (índices de popularidade) e encontre o valor mediano. A mediana é o elemento que estaria no meio da lista se ela fosse ordenada.

- Se N for ímpar, a mediana é o elemento na posição $(N-1) / 2$ da lista ordenada.
- Se N for par, considere a mediana inferior, que também está na posição $(N-1) / 2$ (divisão inteira).

Seu programa deve ser eficiente o suficiente para passar nos limites de tempo, o que significa que uma solução $O(N^2)$ provavelmente falhará. Uma solução com complexidade média $O(N)$ é esperada.

Formato da Entrada:

- A primeira linha conterá um inteiro N , o número de bandas.
- A segunda linha conterá N inteiros separados por espaço, representando os índices de popularidade (valores entre 0 e 100).

Formato da Saída:

- Imprima uma única linha com o valor do índice de popularidade mediano.

Questão 2: Organizador de Documentos

Foco do Algoritmo: Merge Sort

História do Problema:

Um estagiário recebeu uma tarefa aparentemente simples: juntar duas caixas de documentos, A e B, em uma única caixa C. O problema é que os documentos em cada

caixa já estão organizados por data, e a caixa final C também precisa estar perfeitamente ordenada por data. Fazer a junção e depois reordenar tudo seria um desperdício de tempo.

Tarefa:

Escreva um programa que leia duas listas de inteiros já ordenadas e as intercale em uma única lista final ordenada, que deve ser impressa na tela.

Formato da Entrada:

- A primeira linha conterá um inteiro N , o tamanho da primeira lista.
- A segunda linha conterá N inteiros ordenados, separados por espaço.
- A terceira linha conterá um inteiro M , o tamanho da segunda lista.
- A quarta linha conterá M inteiros ordenados, separados por espaço.

Formato da Saída:

- Imprima uma única linha com os $N+M$ elementos das duas listas, de forma intercalada e ordenada, separados por espaço.

Questão 3: Corrida dos Gaps

Foco do Algoritmo: Shell Sort

História do Problema:

Você está participando de uma "corrida de algoritmos" onde o desafio é ordenar um grande volume de números usando o Shell Sort. No entanto, há uma pegadinha: o juiz da competição fornecerá a sequência de "gaps" que você deve usar. Seu programa precisa ser flexível o suficiente para aplicar qualquer sequência de gaps fornecida.

Tarefa:

Escreva um programa que leia um vetor de N inteiros e uma sequência de K gaps. Ordene o vetor de N inteiros usando o algoritmo Shell Sort com a sequência de gaps fornecida.

Formato da Entrada:

- A primeira linha conterá um inteiro N , o tamanho do vetor a ser ordenado.
- A segunda linha conterá N inteiros separados por espaço.
- A terceira linha conterá um inteiro K , o número de gaps na sequência.
- A quarta linha conterá K inteiros, representando a sequência de gaps a ser usada (ex: 7 3 1).

Formato da Saída:

- Imprima uma única linha com o vetor ordenado, com elementos separados por um espaço.

Questão 4: Estoque de Supermercado

Foco do Algoritmo: Bucket Sort

História do Problema:

Um supermercado precisa gerar um relatório ordenando todos os seus produtos por preço, do mais barato ao mais caro. Os preços dos produtos variam entre R\$0.00 e R\$100.00. Como a distribuição de preços é razoavelmente uniforme dentro dessa faixa, um algoritmo de ordenação linear, como o Bucket Sort, pode ser extremamente rápido.

Tarefa:

Escreva um programa em C para ordenar uma lista de N preços de produtos. Os preços são números de ponto flutuante.

Formato da Entrada:

- A primeira linha conterá um inteiro **N**, o número de produtos.
- As próximas **N** linhas conterão cada uma um preço (um número float/double com duas casas decimais, entre 0.01 e 100.00).

Formato da Saída:

- Imprima os **N** preços em ordem crescente, um por linha, formatados com duas casas decimais.

Questão 5: O Mínimo Esforço

Foco do Algoritmo: Selection Sort

História do Problema:

Uma transportadora precisa organizar caixas pesadas em um contêiner. O robô que move as caixas gasta muita energia para cada troca de posição. Para economizar bateria, o supervisor de logística pediu um algoritmo que ordene as caixas por peso, garantindo o menor número de trocas possível.

Tarefa:

Escreva um programa que leia uma lista de N pesos e a ordene usando o algoritmo Selection Sort. Ao final, imprima o número total de trocas (swaps) realizadas. Uma troca é definida como a operação de permutar dois elementos de posição no vetor.

Formato da Entrada:

- A primeira linha conterá um inteiro **N**.
- A segunda linha conterá **N** inteiros separados por espaço, representando os pesos das caixas.

Formato da Saída:

- A primeira linha deve conter o número total de trocas realizadas.
- A segunda linha deve conter o vetor ordenado, com elementos separados por espaço.