

LISTA 1 - CONCEITOS INTRODUTÓRIOS

SEÇÃO 1 - RESPONDA OBJETIVAMENTE CADA UMA DAS PERGUNTAS A SEGUIR:

- a) O que é um algoritmo?

Algoritmo é um conjunto de passos/procedimentos lógicos finitos que definem como determinada tarefa deve ser realizada para solucionar o 'problema'.

- b) Para ordenar 100 números em ordem crescente, qual é a entrada e a saída esperada?

A entrada seria os 100 números e a saída seria os 100 números em ordem crescente.

- c) Uma receita culinária é um algoritmo computacional? Por quê?

*Uma receita culinária **não** é um algoritmo computacional, pois uma receita culinária é um algoritmo de propósito geral que, no caso, é basicamente para cozinhar, não necessitando de uma linguagem de programação... O algoritmo computacional servirá para lidar com tarefas que deixam de ser desempenhadas pelo homem para ser realizada por máquinas/computadores.*

- d) Por que os algoritmos computacionais são independentes de linguagens?

Pois a linguagem é mais uma ferramenta para resolver o problema, o que realmente importa para a resolução é toda a lógica por trás do algoritmo, a linguagem de programação vai ser o seu "facilitador" para a resolução do problema. Consegue-se aplicar um mesmo algoritmo em diferentes linguagens de programação.

- e) Por que os algoritmos computacionais diferem dos programas?

O que difere um algoritmo, que é uma sequência de passos que define como determinada tarefa deve ser realizada para solucionar um problema, de um programa, é que um programa é a representação de um algoritmo em uma linguagem apropriada para instruir um computador a o que fazer com dos dados para transformá-los em informação.

- f) Quais as três formas básicas de representar um algoritmo?

Através de Fluxogramas, Pseudocódigo e Descrição Narrativa.

- g) Qual é a vantagem da representação do algoritmo em pseudocódigo em relação às outras formas básicas de representação de um algoritmo?

O pseudocódigo garante uma vantagem em relação às outras formas porque ele se aproxima da escrita/sintaxe de uma linguagem de programação, ou seja, possui expressões pré-definidas para representar ações e fluxos de controles, sendo vista como uma linguagem de programação simplificada, facilitando a codificação futura.

h) Quais os passos necessários para escrever um bom algoritmo?

Entender o problema, entender entradas e a saída esperada, projetar, pode-se representar o problema em um fluxograma ou pseudocódigo, comentar os passos do algoritmo...

i) O que são estruturas de dados?

Estruturas de dados são maneiras de armazenar e organizar dados para fins de operar sobre eles sendo que diferentes tipos de aplicações utilizam diferentes estruturas de dados.

j) Após a identificação do problema, quais são os passos necessários para escrever um programa?

Planejar, entender a entrada, entender o que se espera na saída, elaborar um método, se o método é correto, analisar a complexidade do método até ela se tornar aceitável.

k) O que é um tipo abstrato de dados e uma modelagem concreta do tipo abstrato de dados? Exemplifique.

Tipo Abstrato de dados é uma especificação de um conjunto de dados e operações que podem ser executadas sobre esses dados, exemplos de TAD básicos e padrão são Listas, Pilhas e Filas. Modelagem concreta do tipo abstrato de dados seria como armazenar o TAD em uma memória e como os algoritmos devem ser usados para implementar as operações, ou seja, a materialização do Tipo Abstrato de Dados, exemplo de Modelagem concreta do TAD seria uma pilha armazenada como lista encadeada ou array.

l) Por que um bom projeto de um algoritmo normalmente leva a uma boa implementação?

Pois todas as ideias principais já iriam ter sido estudadas e a tradução do projeto em programa é quase mecânica.

m) Por que é importante se preocupar com a manutenibilidade e a documentação de um algoritmo em um contexto em que você desenvolve o algoritmo com outras pessoas?

É importante se preocupar pois quanto melhor você escreve o algoritmo e faz uma ótima documentação, a manutenção será feita de maneira melhor e ficará com um entendimento melhor tanto para quem escreveu quanto para as outras pessoas.

n) Por que analisar a eficiência de algoritmos se os computadores estão cada dia mais rápidos?

Pois dependendo da solução que você vai utilizar em determinada situação, ela não será executada em um tempo ideal para tal situação, ou seja, precisa-se analisar a eficiência para resolver de maneira mais ideal/rápida um problema.

o) Para avaliar a eficiência de um algoritmo, o que precisa ser determinado?

Para avaliar a eficiência de um algoritmo é necessário determinar duas complexidades: Complexidade de tempo e complexidade de espaço. A primeira seria quanto tempo é necessário para computar um resultado para uma instância do problema ser resolvido, e a segunda seria quanto espaço de memória é preciso para armazenar a estrutura de dados.

p) A complexidade de tempo de um algoritmo pode ser de pior caso, melhor caso ou caso médio. Qual dessas complexidades é mais importante para o estudo da complexidade de tempo? Por quê?

A mais importante seria a de “Caso médio”, pois é através dele que se consideram todas as possíveis instâncias e mede-se o tempo médio de cada uma.

q) Para avaliar a eficiência de um algoritmo objetivamente, quais os fatores devem ser considerados?

Deve-se considerar: como o programador implementou o algoritmo/estruturas de dados, características do computador (como a velocidade da CPU e a capacidade/ velocidade de acesso à memória primária e secundária) e o Software (Linguagem, Compilador e Sistema Operacional).