

1.

a) O que significa eficiência de tempo de um algoritmo

R: Na utilização dos recursos do computador o tempo de execução não depende somente do algoritmo mas também do hardware.

b) Quais as grandezas físicas que influenciam a eficiência de tempo de um algoritmo na prática.

R: Complexidade temporal e espacial

c) Quais dessas grandezas nós realmente utilizamos para expressar a eficiência de tempo de um algoritmo, utilizando-as para o nosso modelo de computação ?

R: Complexidade temporal pq a espacial nos tempos atuais temos de sobra

d) De quais nós abstraímos e por quê ?

R: Do tempo de execução, não depende somente do algoritmo mas das instruções do computador.

e) Que notação utilizamos na prática para expressar a complexidade de tempo de um algoritmo ?

R: Notação Big O

2) Uma das possíveis formas de se descrever a complexidade de um algoritmo é a chamada Notação O, que é definida da seguinte forma: $T(n) = O(f(n))$ se existem constantes c e n_0 tais que $T(n) \leq c \cdot f(n)$ quando $n > n_0$. Explique o que você entendeu por esta definição?

R: Usamos a Notação Big O para delimitar assintoticamente o crescimento do tempo de execução dentro dos fatores constantes superiores e inferiores sendo executada com complexidade $O(\log n)$ é o tempo de execução no pior caso, e $O(1)$ sendo o tempo de execução no melhor caso, ou seja o tempo de execução da busca binária nunca é pior que $O(\log n)$, mas pode ser melhor em alguns casos.

Se um tempo de execução é $O(f(n))$, então para um n suficientemente grande, o tempo de execução é no máximo $k \cdot f(n)$ para uma constante k .

3)

a) Explique que tipos de problemas ou algoritmos costumam ter complexidade da ordem de $n \log n$ como os identificamos.

R: São algoritmos que dividem o problema em problemas menores e depois juntam as soluções, por exemplo HeapSort, MergeSort, QuickSort

b) Quais problemas que possuem geralmente complexidade da ordem de $\log n$? Cite dois exemplos:

R: Geralmente são problemas que possuem um laço de repetição e condicionais dentro desse laço como a Busca Binária onde em função de N realiza comparações.

c) Quais os problemas que costumam ser exponenciais? Cite um exemplo:

R: Problemas que possuem laços como `for` ou `while`, por exemplo multiplicação de matrizes,

4-

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main(){
    int n, i, media, avaliacao, soma=0;
    printf("Digite a quantidade de notas: ");
    scanf("%d", &n);
    for(i=0; i<n; i++){
        printf("Digite a nota do aluno %d: ", i+1);
        scanf("%d", &avaliacao);
        soma = soma + avaliacao;
    }
    media = soma/n;

    printf("Media final: %d", media);
}
```

5.as operações primitivas que vem ocorrendo no algoritmo acima é de $\text{soma} = \text{soma} + \text{avaliacao}$ que vai pegar e vai somar todas as notas cada vez que um usuário informar e a outra é $\text{media} = \text{soma}/n$ que será feito a média do aluno aritmeticamente

6-

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main(){
    int tamanho;
    printf("Digite o tamanho do vetor: ");
    scanf("%d", &tamanho);
    int vet[tamanho], numero, achou=0, i;
    for(i=0; i<tamanho; i++){
        printf("Digite um numero para a posição %d: ", i);
        scanf("%d", &vet[i]);
    }
    printf("Digite o valor que deseja buscar: ");
```

```
scanf("%d", &numero);
for(i=0; i<tamanhoo; i++){
    if(numero == vet[i]){
        achou=1;
    }
}
if(achou == 1)
    printf("achou\n");
else
    printf("nao achou\n");
}
```

7- A operação primitiva vem a ser o tamanho do vetor informado e procurando um número para posição e seus laços de for que vem fazer as operações