Lista 04 – Estrutura de Dados -
Alocação de memória e Complexidade de Algoritmo

1) Expresse a função $n^3/1000 – 100n^2 – 100n +3$ em termos da notação Θ

2) Análise a complexidade dos seguintes algoritmos, referindo o melhor caso, caso médio e o pior caso.

a)

```
void func1(){
    int i,j,k, soma;
    for(i=1; i<n-1; i++){
        for(j=i+1; j<n; j++){
            for(k=1; k<j;k++){
                soma=1
}}}}
```

b)

```
float f(int M[], int N[], int P){
    int i,j;
    float c=1.0;
    for(i=1; i<c; i++){
        if([i] > 1000){
            for(j=P-1; j>=0; j--){
                c+= M[i]*N[j];
            }}
        else if (M[i] < 500){
            for(j=P; j<P*P; j+=2){
                c += M[i]*N[j];
            }}
        else{
            for(j=1; j<P; j=2*){
                c += M[i]*N[j];
}}}}
```

3) Faça um programa de estoque de loja:

a) Crie uma struct PRODUTO que possua um CODIGO(inteiro), uma NOME(char) e um PRECO (real). Ao criar esse novo tipo, crie variáveis ponteiros para essa estrutura.
b) Usando alocação de memória, crie um vetor de 5 elementos, porém criando 1 por 1 dos elementos do vetor. Faça o cadastro de 5 produtos.
c) Ordene os produtos de acordo com o preço do produto. Ordene de forma decrescente.
d) Imprima a média do valor dos preço dos produtos.
e) d) Imprima todos os produtos.

4) O programa deve criar uma matriz quadrática (dinamicamente) de valores zeros.

a) Perguntar ao usuário que tamanho será essa matriz. Utilizar um ponteiro para indicar o tamanho dessa matriz.
b) alocar em tempo de execução (dinamicamente) uma matriz quadrada de zeros, passando como parâmetros o número de linhas e colunas.
c) Inclua a biblioteca ctime e utilize a função clock para determinar o tempo de execução da alocação de memória com o tamanho da matriz quadrada de tamanho 10 e 1000000.

5) Faça um programa que leia um vetor de inteiros do usuário, remova valores consecutivos repetidos e mostra o vetor atualizado. O programa deve alocar memória para o vetor inicial e usar a função realloc para ajustar o tamanho do vetor. O tamanho do vetor deve ser sempre o mínimo necessário para armazenar os valores, ou seja, usar malloc de 1 em 1. O programa deve mostrar o tamanho final do vetor, como no exemplo abaixo.

Tamanho inicial = 10

0	0	1	1	1	2	2	3	0	3

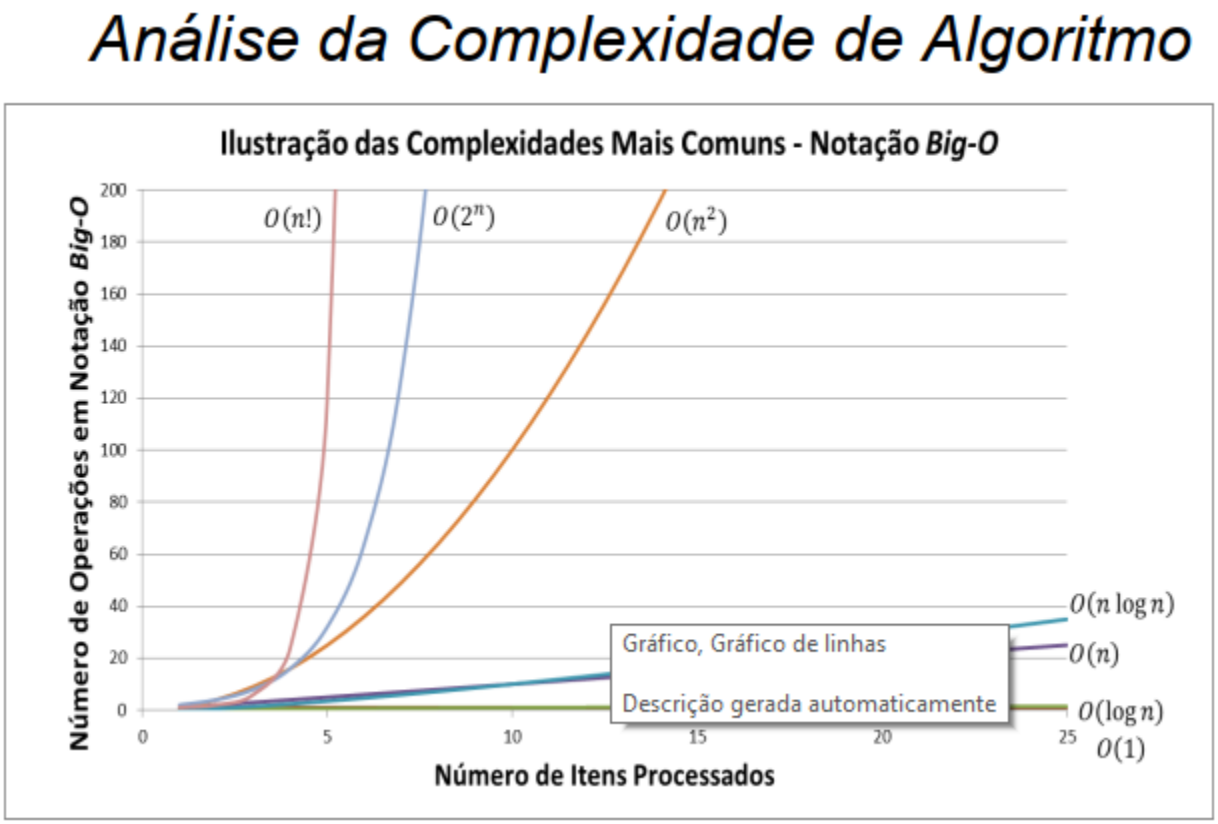
Tamanho final = 6

0	1	2	3	0	3

6) Faça o exercício 4 usando duas vezes o malloc para criar a matriz, porém agora a matriz pode ser m x n, ou seja, não quadrática. Procure na internet a alocação dinâmica de memória para matrizes.

7) Crie um programa simples que exiba o fatorial de um número e calcule o tempo de execução do programa usando a biblioteca ctime e a função clock. Peça para o usuário dar o número que deseja fazer fatorial. Faça a análise da complexidade desse código, escreva como comentário no final.

8) Explique a imagem abaixo, o que é, para que serve, em que momento é utilizado e dê um exemplo de aplicação prática do dia a dia que pode ser utilizado.



9) Quais fatores determinam o custo final de um algoritmo?

10) Quando um código tem uma PA ou uma PG qual a complexidade do algoritmo? Que tipos de códigos que aparecem uma PA ou PG?