

# Trabalho Prático: Busca e Ordenação

## Faculdade UnB Gama - FGA

### Universidade de Brasília - UnB

Aplicação de Algoritmos de Busca e Ordenação em Registros de  
Operações de Vendas da Amazon.com

Disciplina: FGA0147 - Estruturas de Dados e Algoritmos  
Prof. Dr. Nilton Correia da Silva

Grupo G:  
Esther Silva Cardoso de Sousa - 190106034  
Ian Lucca Soares Mesquita - 211045140  
Leticia Torres Soares Martins - 202016702  
Marcos Santos Bittar - 200023748  
Otávio Henrique Moraes Brito - 211043692

## 1. Relatório Técnico

### 1.1. Objetivo

Neste trabalho prático desenvolvemos um programa por meio da ordenação e busca aprendido na aula, com o objetivo de ordenar um dataset do registro de vendas da empresa da Amazon. Nele foi informado dois campos: Categoria e Valor, e a partir dele, ordenar pela lucratividade dos comércios eletrônicos. Nós construímos o código a partir da ordenação **Insertion Sort**.

:

### 1.2. Programa

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

//Struct que recebe os dados do arquivo original
struct dados{
    char category[40];
    float amount;
};

//InsertionSort que realiza a ordenação do ammount
void insertionSort(float list[], int n,int o[]){
    int i, j;
    float next;
    for (i=1; i<n; i++) {
        next = list[i];
        for (j=i-1; j>=0&&next>list[j];j--){
            list[j+1] = list[j];
            o[i]=j;//Contagem do indice, relacionando a linha antiga e a linha nova
        }
        list[j+1] = next;
    }
}

//Acessa uma linha determinada do arquivo
FILE acessLine(FILE *fp, int line){
    char ch;
    int count = 0;
    while(count<line){
        ch = fgetc(fp);
        if(ch == '\n'){
```

```

        count = count+1;
    }
}
return *fp;
}

int main(){
    //Chamada de structs, variaveis, ponteiros e arquivos
    struct dados D;
    float lis[1290];
    int line,j,o[1290], num=0;
    FILE *fp, *fp2;
    FILE *copy;
    //Abertura do arquivo e checagem se ele realmente foi criado
    fp = fopen ( "Trimmed_File.csv", "r" ) ;
    if ( fp == NULL ){
        puts ( "Cannot open file" ) ;
        system("pause");
        exit(0) ;
    }
    //Leitura do arquivo e entrada desses dados na struct
    while(1){
        char result;
        result = fscanf(fp, "%s %f", D.category, &D.amount);
        lis[num]=D.amount;
        num=num+1;
        if ( result == EOF)
            break;
        //printf("%s - %.2f \n", D.category,D.amount);
    }
    fclose(fp);
    //Chamada para o insertionsort
        //copy = fopen("Amazonsalecopy.csv","w");
    insertionSort(lis,num,o);
    printf("\n ORDENADO: \n");
    //Abertura do arquivo de saida e checagem desse
    fp2 = fopen("Arquivo_Ordenado.csv", "w");
    if ( fp2 == NULL ){
        puts ( "Cannot open file" ) ;
        system("pause");
        exit(0) ;
    }
}

```

//For que printar os valores ordenados os associando as categorys respectivas

```
for(j=0;j<=num;j++){
    fp = fopen ( "Amazonsale.csv", "r" ) ;
    line=o[j];
    *fp = acessLine(fp, line);
    char ch = fgetc(fp);

    while(ch !=' '){
        fputc(ch,fp2);
        ch = fgetc(fp);
    }

    //printf("%d\n",o[j]);
    //printf("%.2f\n",lis[j]);
    fprintf(fp2," %.2f\n",lis[j]);

}
printf("estou aqui");
//fclose(copy);
fclose(fp2);
fclose(fp);
}
```

### 1.3. Complexidade do Algoritmo de Ordenação

O programa teve sua complexidade definida como o pior caso sendo  $O(n^2)$  e o melhor caso sendo  $O(n)$ .

### 1.4. Tempo de Processamento

O tempo de processamento do programa variou entre os computadores da equipe, todavia o grupo chegou na média dos tempos sendo ela **180 segundos**. Utilizando um computador com windows 10/11, com um processador intel i7 décima geração.

### 1.5. Conclusão

No geral, o grupo não encontrou dificuldades para cooperar entre si durante a realização do trabalho, além de conseguir manter um bom ritmo de trabalho.

# Link GitHub do Trabalho

<https://github.com/lanLucca12/TrabalhoFinal-EDA>