

Lista de exercícios Estrutura de Dados 1

Sthefany Sther Pequeno Alves – RGM: 1723535547

Departamento de Ciências da computação – Turma: 3A – Universidade Cidade de São Paulo (UNICID) – São Paulo – SP – Brás

sthefanypequeno@gmail.com

Abstract. These exercises describe the second list of data structures, in which it aims to reinforce the studies. The exercises are at the base of the C language, whose codifications are described and the activities in its great majority are to improve the use of the methods of insertion and selection in the codes. This article is in the proposed standards SBC (Brazilian Computer Society).

Resumo. Estes exercícios descrevem a segunda lista de estrutura de dados, na qual tem a sua finalidade de reforçar os estudos. Os exercícios estão na base da linguagem C, cujo as codificações são descritas e as atividades em sua grande maioria são para aprimorar o uso dos métodos de inserção e seleção nos códigos. Este artigo, está nas normas propostas SBC (Sociedade Brasileira de Computação).

1. Escreva uma função que reorganize um vetor $v[0..n-1]$ de modo que ele fique em ordem estritamente crescente.

Codificação:

```
# include <stdio.h>

# include <locale.h>

# include <stdbool.h>

#define NYA 8

int vet[NYA]={1,2,4,5,3,8,-6,9}, x, c;

void imprimir ();

void insertSort (int v[NYA], int n);

void main (void) {

    setlocale (LC_ALL, "Portuguese");

    printf ("~~~~ Exercício 1 ~~~~");

    printf ("\nVetor na ordem que foi apresentada: " );
```

```

        imprimir ();
        printf("\n");
        insertSort (vet,NYA);
        printf("\nDepois de ter executado o código: " );
        imprimir ();
    }
void imprimir(){
    int s;
    for (s = 0; s < NYA; s++) printf ("%d\t\t", vet[s]);
}
void insertSort(int v[NYA],int n)
{
    int j;
    for (j = 1; j < n; ++j) {
        x = v[j];
        int i;
        for (i = j-1; i >= 0 && v[i] > x; --i){
            v[i+1] = v[i];
            c++;
            printf("Iteração %d:\t",c);
            imprimir();printf("\tx = %d\n",x);
        }
        v[i+1] = x;
    }
}
}

```

```

C:\Users\user\Documents\TrabalhoLista2.exe
~~~~~ Exercício 1 ~~~~~
Vetor na ordem que foi apresentada: 1 - 2 - 4 - 5 - 3 - 8 - -6 - 9 -
Iteração 1: 1 - 2 - 4 - 5 - 8 - -6 - 9 - x = 3
Iteração 2: 1 - 2 - 4 - 5 - 8 - -6 - 9 - x = 3
Iteração 3: 1 - 2 - 3 - 4 - 5 - 8 - 9 - x = -6
Iteração 4: 1 - 2 - 3 - 4 - 5 - 8 - 9 - x = -6
Iteração 5: 1 - 2 - 3 - 4 - 5 - 8 - 9 - x = -6
Iteração 6: 1 - 2 - 3 - 4 - 5 - 8 - 9 - x = -6
Iteração 7: 1 - 2 - 3 - 4 - 5 - 8 - 9 - x = -6
Iteração 8: 1 - 1 - 2 - 3 - 4 - 5 - 8 - 9 - x = -6

Depois de ter executado o código: -6 - 1 - 2 - 3 - 4 - 5 - 8 - 9 -
-----
Process exited after 0.03883 seconds with return value 4
Pressione qualquer tecla para continuar. . .

```

Figura 1. Resultado do código de execução do exercício 1.

2. Crie uma função que conte quantas iterações ocorrem ao ordenar um vetor.

Codificação:

```

#include <stdio.h>

#include <locale.h>

#include <stdbool.h>

#define NYA 8

int vet[NYA]={1,2,4,5,3,8,-6,9}, x, c;

void imprimir ();

void insertSort (int v[NYA], int n);

void qtditeracao ();

void main (void) {

    setlocale (LC_ALL, "Portuguese");

    printf ("~~~~~ Exercício 2 ~~~~~");

    printf ("\nVetor na ordem que foi apresentada: ");

    imprimir ();

    printf("\n");

    insertSort (vet,NYA);

    printf("\nDepois de ter executado o código: ");

    imprimir ();

    printf("\n \n");

    qtditeracao ();

}

void imprimir(){

```

```

    int s;

    for (s = 0; s < NYA; s++) printf ("%d\t\t", vet[s]);
}

void insertSort(int v[NYA],int n)
{
    int j;

    for (j = 1; j < n; ++j) {
        x = v[j];

        int i;

        for (i = j-1; i >= 0 && v[i] > x; --i){
            v[i+1] = v[i];

            c++;

            printf("Iteração %d:\t",c);

            imprimir();printf("\tx = %d\n",x);
        }

        v[i+1] = x;
    }
}

void qtditeracao () {
    printf("Teve %d iterações", c)
}

```

```

Exercício 2
Vetor na ordem que foi apresentada: 1 2 4 5 3 8 -6 9
Iteração 1: 1 2 4 5 8 -6 9 x = 3
Iteração 2: 1 2 4 5 8 -6 9 x = 3
Iteração 3: 1 2 3 4 5 8 9 x = -6
Iteração 4: 1 2 3 4 5 8 9 x = -6
Iteração 5: 1 2 3 4 5 8 9 x = -6
Iteração 6: 1 2 3 4 5 8 9 x = -6
Iteração 7: 1 2 3 4 5 8 9 x = -6
Iteração 8: 1 1 2 3 4 5 8 9 x = -6

Depois de ter executado o código: -6 1 2 3 4 5 8 9
Teve 8 iterações
Process exited after 0.03847 seconds with return value 16
Pressione qualquer tecla para continuar. . .

```

Figura 2. Resultado do código de execução do exercício 2.

3. Busca binária. O papel do for interno na função inserção é encontrar o ponto onde $v[j]$ deve ser inserido em $v[0..j-1]$. Considere fazer isso com uma busca binária. Analise o resultado.

Codificação:

```
# include <stdio.h>

# include <locale.h>

# include <stdbool.h>

#define NYA 8

int vet[NYA]={1,2,4,5,3,8,-6,9}, x, c, bb;

void imprimir ();

void insertSort (int v[NYA], int n);

void buscaBinaria();

void main (void) {

    setlocale (LC_ALL, "Portuguese");

    printf ("~~~~ Exercício 3 ~~~~");

    printf ("\nVetor na ordem que foi apresentada: " );

    imprimir ();

    printf("\n");

    insertSort (vet,NYA);

    printf("\nDepois de ter executado o código: " );

    imprimir ();

    printf ("\n");

    buscaBinaria();

}

void imprimir(){

    int s;

    for (s = 0; s < NYA; s++) printf ("%d\t\t", vet[s]);

}
```

```

void insertSort(int v[NYA],int n)
{
    int j;
    for (j = 1; j < n; ++j) {
        x = v[j];
        int i;
        for (i = j-1; i >= 0 && v[i] > x; --i){
            v[i+1] = v[i];
            c++;
            printf(" Iteração %d:\t",c);
            imprimir();printf("\tx = %d\n",x);
        }
        v[i+1] = x;
    }
}

void buscaBinaria(){
    printf(" Insira um número para verificar se ele esta no vetor\n");
    scanf("%d",&bb);
    int direita, esquerda, meio;
    bool encontrado = 0;
    esquerda = 0;
    direita = NYA - 1;
    while(esquerda<=direita && !encontrado){
        meio=(direita+esquerda)/2;
        printf("\n Meio do vetor =\tv[vetor[%d] = %d",meio,vet[meio]);
        if (vet[meio] == bb) encontrado = 1;
        else if (bb < vet[meio]) direita = meio - 1;
        else esquerda = meio + 1;
    }
}

```

```

printf("\n\n");

if(encontrado) printf(" O número digitado: %d \n é pertencente ao vetor e está na
posição %d.\n\n",bb ,meio);

else printf(" O número não pertence ao vetor.\n\n");

}

```

```

~~~~~ Exercício 3 ~~~~~
Vetor na ordem que foi apresentada: 1 - 2 - 4 - 5 - 3 - 8 - -6 - 9 -
Iteração 1: 1 - 2 - 4 - 5 - 5 - 8 - -6 - 9 - x = 3
Iteração 2: 1 - 2 - 4 - 5 - 5 - 8 - -6 - 9 - x = 3
Iteração 3: 1 - 2 - 3 - 4 - 5 - 8 - 8 - 9 - x = -6
Iteração 4: 1 - 2 - 3 - 4 - 5 - 5 - 8 - 9 - x = -6
Iteração 5: 1 - 2 - 3 - 4 - 4 - 5 - 8 - 9 - x = -6
Iteração 6: 1 - 2 - 3 - 3 - 4 - 5 - 8 - 9 - x = -6
Iteração 7: 1 - 2 - 2 - 3 - 4 - 5 - 8 - 9 - x = -6
Iteração 8: 1 - 1 - 2 - 3 - 4 - 5 - 8 - 9 - x = -6

Depois de ter executado o código: -6 - 1 - 2 - 3 - 4 - 5 - 8 - 9 -
Insira um número para verificar se ele esta no vetor
5

Meio do vetor = vetor[3] = 3
Meio do vetor = vetor[5] = 5

O número digitado: 5
é pertencente ao vetor e está na posição 5.

-----
Process exited after 7.431 seconds with return value 69
Pressione qualquer tecla para continuar. . .

```

Figura 3. Resultado da codificação do exercício 3.

```

~~~~~ Exercício 3 ~~~~~
Vetor na ordem que foi apresentada: 1 - 2 - 4 - 5 - 3 - 8 - -6 - 9 -
Iteração 1: 1 - 2 - 4 - 5 - 5 - 8 - -6 - 9 - x = 3
Iteração 2: 1 - 2 - 4 - 5 - 5 - 8 - -6 - 9 - x = 3
Iteração 3: 1 - 2 - 3 - 4 - 5 - 8 - 8 - 9 - x = -6
Iteração 4: 1 - 2 - 3 - 4 - 5 - 5 - 8 - 9 - x = -6
Iteração 5: 1 - 2 - 3 - 4 - 4 - 5 - 8 - 9 - x = -6
Iteração 6: 1 - 2 - 3 - 3 - 4 - 5 - 8 - 9 - x = -6
Iteração 7: 1 - 2 - 2 - 3 - 4 - 5 - 8 - 9 - x = -6
Iteração 8: 1 - 1 - 2 - 3 - 4 - 5 - 8 - 9 - x = -6

Depois de ter executado o código: -6 - 1 - 2 - 3 - 4 - 5 - 8 - 9 -
Insira um número para verificar se ele esta no vetor
20

Meio do vetor = vetor[3] = 3
Meio do vetor = vetor[5] = 5
Meio do vetor = vetor[6] = 8
Meio do vetor = vetor[7] = 9

O número não pertence ao vetor.

-----
Process exited after 2.219 seconds with return value 0
Pressione qualquer tecla para continuar. . .

```

Figura 4. Resultado da codificação do exercício 3.

Descrição de aprendizagem:

Apreendi bastante sobre inserção e seleção, tive algumas dificuldades no exercício 1, mas meus colegas de classe e as videoaulas suprimiram todas elas, e com isso consegui desenvolver melhor os exercícios 2 e 3 de forma simples e prática, obtendo melhor resultado a cada tentativa.

Refêrencias:

Juliano Ratusznei (2021) – Estrutura de dados 1 – Apostila Ordenação;

Juliano Ratusznei (2021) – Estrutura de dados 1 – Blackboard – Videoaula 09/04;