

UERJ – CAMPUS ZONA OESTE CIÊNCIA DA COMPUTAÇÃO

**RUAN SOARES DA SILVA FONSECA - 1923333070
GUILHERME DE OLIVEIRA VIEIRA BISPO - 2013332005
DANYLO HENRIQUE DA SILVA DOS SANTOS - 2013332021**

**TRABALHO AV2 – ESTRUTURA DE DADOS 1
PROF. DR. DENIS GONÇALVES COUPLE**

**RIO DE JANEIRO
2023**

RUAN SOARES DA SILVA FONSECA - 1923333070
GUILHERME DE OLIVEIRA VIEIRA BISPO - 2013332005
DANYLO HENRIQUE DA SILVA DOS SANTOS - 2013332021

ESTRUTURA DE DADOS
TRABALHO COMPLEMENTAR PARA AV2

Trabalho acadêmico apresentado à disciplina de Estrutura de Dados 1 do Curso de Ciência da Computação como requisito de nota parcial da AV2. Requerido pelo prof. Dr.Denis Golçalves Couple.

RIO DE JANEIRO
2023

SUMÁRIO

1	INTRODUÇÃO	4
2	QUESTÃO 1	5
2.1	CÓDIGO DA QUESTÃO 1	5
2.2	PRINTS DA QUESTÃO 1	6
3	QUESTÃO 2	6
3.1	CÓDIGO DA QUESTÃO 2	6
3.2	PRINTS DA QUESTÃO 2	7
4	QUESTÃO 3	8
4.1	CÓDIGO DA QUESTÃO 3	8
4.2	PRINTS DA QUESTÃO 3	10
5	CONCLUSÃO.....	10
6	BIBLIOGRAFIA.....	12

1 INTRODUÇÃO

Foram desenvolvidos 3 programas em C e C++ para este trabalho, cada um para sua respectiva questão, utilizando técnicas de algoritmos gulosos e programação dinâmica aprendidas sem sala de aula.

Nesta documentação apresentaremos os códigos dos sistemas e os prints da execução com os casos possíveis.

2 QUESTÃO 1

2.1 CÓDIGO DA QUESTÃO 1

A seguir o código para implementação da questão 1, sobre o sistema de viagem com a menor quantidade possível:

```
#include <stdio.h>
#include <iostream>
#include <locale.h>
#define MAX_POSTOS 100

using namespace std;

int main()
{
    int n, m, k, i;

    cout << "numero de postos na estrada: ";
    cin >> k;
    cout << "Tamanho da estrada (km): ";
    cin >> m;
    cout << "Capacidade do carro (km): ";
    cin >> n;

    cout << endl;
    int* p = new int[k];
    for (i = 0; i < k; i++)
    {
        cout << "posição do posto " << i << " : ";
        cin >> p[i];
    }

    int paradaAtual = 0;
    int combustivel = n;
    int paradas = 0;

    for (i = 0; i < k; i++)
    {
        if (p[i] - paradaAtual > combustivel)
        {
            paradas++;
            combustivel = n;
        }

        combustivel -= p[i] - paradaAtual;
        paradaAtual = p[i];
    }

    if (m - paradaAtual > combustivel)
    {
        paradas++;
    }

    if (n > m)
        cout << endl << "Concluiu sem fazer paradas";

    cout << endl << "paradas -> " << paradas << endl;
```

```

        delete[] p;
    }

```

2.2 PRINTS DA QUESTÃO 1

A seguir os prints da execução do programa:

```

6.9 Console de Depuração do Microsoft Visual Studio
numero de postos na estrada: 3
Tamanho da estrada (km): 100
Capacidade do carro (km): 40

posicao do posto 0 : 30
posicao do posto 1 : 60
posicao do posto 2 : 90

paradas -> 2

O C:\Users\User\Desktop\faculdade\6 periodo\Estrutura de Dados 1\av2\av2 ed1 q1\x64\Debug\av2 ed1 q1.exe (processo 23676) foi encerrado com o código 0.
Pressione qualquer tecla para fechar esta janela...

```

```

6.9 Console de Depuração do Microsoft Visual Studio
numero de postos na estrada: 2
Tamanho da estrada (km): 50
Capacidade do carro (km): 60

posicao do posto 0 : 14
posicao do posto 1 : 36

Concluiu sem fazer paradas
paradas -> 0

O C:\Users\User\Desktop\faculdade\6 periodo\Estrutura de Dados 1\av2\av2 ed1 q1\x64\Debug\av2 ed1 q1.exe (processo 19436) foi encerrado com o código 0.
Pressione qualquer tecla para fechar esta janela...

```

3 QUESTÃO 2

3.1 CÓDIGO DA QUESTÃO 2

A seguir o código da questão 2, do sistema para organizar a realização de tarefas visando a menor multa possível:

```

#include<stdio.h>
#include <stdlib.h>
#include <algorithm>
#include <iostream>

#define MAX_TAREFAS 100

using namespace std;

struct Tarefa {
    int dia;
    int multa;
};

bool compara(Tarefa a, Tarefa b) {

```

```

        if (a.dia == b.dia)
        {
            return a.multa > b.multa;
        }
        return a.dia < b.dia;
    }

int main() {
    int n, i;
    cout << "Numero de tarefas: ";
    cin >> n;
    Tarefa* tarefas= new Tarefa[n];

    for (i = 0; i < n; i++) {
        cout << endl;
        cout << "Prazo da tarefa " << i + 1 << " (dias): ";
        cin >> tarefas[i].dia;
        cout << "multa da tarefa " << i + 1 << ": ";
        cin >> tarefas[i].multa;
    }

    sort(tarefas, tarefas + n, compara);

    int dias_trabalhados = 1, multa = 0;
    for (i = 0; i < n; i++) {


        if (dias_trabalhados <= tarefas[i].dia)
        {
            dias_trabalhados += 1;
        }
        else {
            multa += (dias_trabalhados - tarefas[i].dia) * tarefas[i].multa;
            dias_trabalhados += 1;
        }
    }

    printf("\nMulta total: %d\n", multa);
    return 0;
}

```

3.2 PRINTS DA QUESTÃO 2

A seguir os prints da segunda execução:



```

Console de Depuração do Microsoft Visual Studio
Numero de tarefas: 3

Prazo da tarefa 1 (dias): 1
multa da tarefa 1: 10

Prazo da tarefa 2 (dias): 1
multa da tarefa 2: 15

Prazo da tarefa 3 (dias): 3
multa da tarefa 3: 20

Multa total: 10

O C:\Users\User\Desktop\faculdade\6 periodo\Estrutura de Dados 1\av2\av2 ed1 q2\x64\Debug\av2 ed1 q2.exe (processo 20924) foi encerrado com o código 0.
Pressione qualquer tecla para fechar esta janela...

```

Console de Depuração do Microsoft Visual Studio

Numero de tarefas: 4

Prazo da tarefa 1 (dias): 1
multa da tarefa 1: 20

Prazo da tarefa 2 (dias): 1
multa da tarefa 2: 30

Prazo da tarefa 3 (dias): 3
multa da tarefa 3: 20

Prazo da tarefa 4 (dias): 3
multa da tarefa 4: 10

Multa total: 30

O C:\Users\User\Desktop\faculdade\6 periodo\Estrutura de Dados 1\av2\av2 ed1 q2\x64\Debug\av2 ed1 q2.exe (processo 21276) foi encerrado com o código 0.
Pressione qualquer tecla para fechar esta janela...

Console de Depuração do Microsoft Visual Studio

Numero de tarefas: 5

Prazo da tarefa 1 (dias): 3
multa da tarefa 1: 40

Prazo da tarefa 2 (dias): 1
multa da tarefa 2: 20

Prazo da tarefa 3 (dias): 3
multa da tarefa 3: 80

Prazo da tarefa 4 (dias): 2
multa da tarefa 4: 10

Prazo da tarefa 5 (dias): 5
multa da tarefa 5: 100

Multa total: 40

O C:\Users\User\Desktop\faculdade\6 periodo\Estrutura de Dados 1\av2\av2 ed1 q2\x64\Debug\av2 ed1 q2.exe (processo 16364) foi encerrado com o código 0.
Pressione qualquer tecla para fechar esta janela...

4 QUESTÃO 3

4.1 CÓDIGO DA QUESTÃO 3

A seguir o código da questão 3, para o programa que apresenta a menor quantidade de moedas que represente um valor:

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

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

```
#include <locale.h>
```

```
int moedas[] = {2, 5, 10, 25, 50};
```

```
int n = 5;
```

```
void apresentar_moedas(int valor) {
```

```
    int i;
```



```

for (i = n-1; i >= 0; i--) {
    if (valor >= moedas[i]) {
        int num = valor / moedas[i];
        valor = valor % moedas[i];
        puts("\n");
        printf("%d moeda(s) de R$ %d centavos\n", num, moedas[i]);
    }
}

int main() {
    int valor;
    setlocale(LC_ALL, "Portuguese");

    printf("Informe o valor em centavos: ");
    scanf("%d", &valor);
    puts("\n");
    if (valor == 0) {
        printf("Não é possível formar o valor com as moedas disponíveis\n");
    } else {
        printf("A menor quantidade de moedas é:\n");
        apresentar_moedas(valor);
    }
    return 0;
}

```

```
}
```

4.2 PRINTS DA QUESTÃO 3

A seguir os prints da execução do programa para a questão 3:

```
Console de Depuração do Microsoft Visual Studio
Informe o valor em centavos: 15

A menor quantidade de moedas é:
1 moeda(s) de R$ 10 centavos
1 moeda(s) de R$ 5 centavos

O C:\Users\User\Desktop\faculdade\6 periodo\Estrutura de Dados 1\av2\av2 ed1 q3\x64\Debug\av2 ed1 q3.exe (processo 24332) foi encerrado com o código 0.
Pressione qualquer tecla para fechar esta janela...
```

```
Console de Depuração do Microsoft Visual Studio
Informe o valor em centavos: 34

A menor quantidade de moedas é:
1 moeda(s) de R$ 25 centavos
1 moeda(s) de R$ 5 centavos
2 moeda(s) de R$ 2 centavos

O C:\Users\User\Desktop\faculdade\6 periodo\Estrutura de Dados 1\av2\av2 ed1 q3\x64\Debug\av2 ed1 q3.exe (processo 21200) foi encerrado com o código 0.
Pressione qualquer tecla para fechar esta janela...
```

```
Console de Depuração do Microsoft Visual Studio
Informe o valor em centavos: 3

Não é possível formar o valor com as moedas disponíveis

O C:\Users\User\Desktop\faculdade\6 periodo\Estrutura de Dados 1\av2\av2 ed1 q3\x64\Debug\av2 ed1 q3.exe (processo 18068) foi encerrado com o código 0.
Pressione qualquer tecla para fechar esta janela...
```

5 CONCLUSÃO

A programação gulosa e a dinâmica são muito úteis para solucionar problemas, podendo ser melhores do que diversos mecanismos de programação, como por exemplo a de recursividade. Os algoritmos gulosos visam a melhor combinação

disponível em um determinado momento do programa, além de ter uma fácil implementação, apesar de nem sempre resultar na solução ideal. Já a programação dinâmica visa resolver um problema da forma mais eficiente possível e com a melhor solução global.

6 BIBLIOGRAFIA

<https://www.programiz.com/dsa/dynamic-programming>

<https://www.geeksforgeeks.org/greedy-algorithms/>

https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/dynamic-programming.html