



Aluno: Mateus Terra Tavares Ramos **Curso:** Engenharia de Computação **Data:** 24/03/23

Disciplina: Projeto e análise de algoritmos **Prof^a:** Philippe Leal

1)

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

```
#define MAX_ALUNOS 100
```

```
typedef struct {
    int matricula;
    char nome[30];
    float nota1, nota2, media;
} Aluno;
```

```
void preenche_vetor(Aluno *alunos, int *qtd_alunos) {
    printf("Preenchimento do vetor de alunos:\n");
    printf("Informe o numero de alunos (maximo %d): ", MAX_ALUNOS);
    scanf("%d", qtd_alunos);
```

```
    for (int i = 0; i < *qtd_alunos; i++) {
        printf("\nAluno %d:\n", i+1);
        printf("Matricula: ");
        scanf("%d", &alunos[i].matricula);
        printf("Nome: ");
        scanf("%[^\n]s", alunos[i].nome);
        printf("Nota 1: ");
        scanf("%f", &alunos[i].nota1);
        printf("Nota 2: ");
        scanf("%f", &alunos[i].nota2);
        alunos[i].media = (alunos[i].nota1 + alunos[i].nota2) / 2.0;
    }
}
```

```
int busca_binaria_recursiva(Aluno *alunos, int inicio, int fim, char *nome) {
    if (inicio > fim) {
        return -1;
    }
```

```
    int meio = (inicio + fim) / 2;
```

```
    if (strcmp(alunos[meio].nome, nome) == 0) {
        return meio;
    }
```

```
}

if (strcmp(alunos[meio].nome, nome) < 0) {
    return busca_binaria_recursiva(alunos, meio+1, fim, nome);
} else {
    return busca_binaria_recursiva(alunos, inicio, meio-1, nome);
}
}

int main() {
    Aluno alunos[MAX_ALUNOS];
    int qtd_alunos;

    preenche_vetor(alunos, &qtd_alunos);

    char nome[30];
    printf("\nInforme o nome a ser buscado: ");
    scanf("%[^\n]s", nome);

    int posicao = busca_binaria_recursiva(alunos, 0, qtd_alunos-1, nome);

    if (posicao == -1) {
        printf("Nenhum aluno encontrado com o nome %s.\n", nome);
    } else {
        printf("Aluno encontrado:\n");
        printf("Matricula: %d\n", alunos[posicao].matricula);
        printf("Nome: %s\n", alunos[posicao].nome);
        printf("Nota 1: %.1f\n", alunos[posicao].nota1);
        printf("Nota 2: %.1f\n", alunos[posicao].nota2);
        printf("Media: %.1f\n", alunos[posicao].media);
    }

    return 0;
}
```

2)

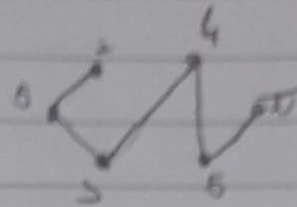
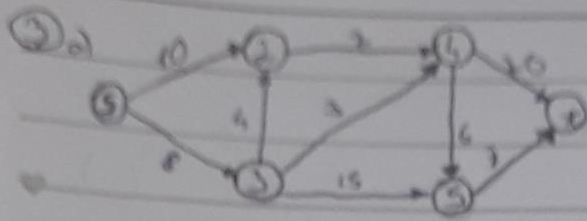
$$2) \begin{cases} T(1) = 1, & n = 1 \\ T(n) = T\left(\frac{n}{2}\right) + O(1), & n > 1 \end{cases}$$

$$\begin{aligned} T(n) &= T\left(\frac{n}{2}\right) + 1 \\ &= T\left(\frac{n}{4}\right) + 1 + 1 \\ &= T\left(\frac{n}{8}\right) + 1 + 1 + 1 \end{aligned} \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} = T\left(\frac{n}{2^k}\right) + k$$

$$\frac{n}{2^k} \rightarrow 2^k = n \rightarrow \log_2(2^k) = \log_2(n) \rightarrow k = \log_2(n)$$

$$T(n) = T(1) + \log_2(n) \quad O(\log n)$$

3)



$$Q = \{\$, \cancel{2}, \cancel{3}, 4, 5, \cancel{6}, \cancel{7}\}$$

$$S = \{5, 3, 2, 4, 5, \cancel{1}\}$$

$$S \rightarrow 2: \infty \cancel{10} \text{ (10)}$$

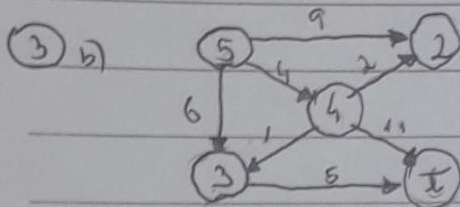
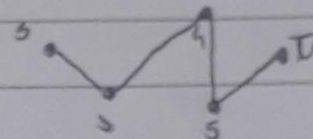
$$S \rightarrow 3: \infty \text{ (8)}$$

$$S \rightarrow 4: \infty \cancel{11} \text{ (11)}$$

$$S \rightarrow 5: \infty \cancel{23} \cancel{15} \text{ (19)}$$

$$S \rightarrow \cancel{1}: \infty \infty \infty \cancel{31} \text{ (24)}$$

memor caminho:



$$Q = \{\$, \cancel{1}, \cancel{3}, 4, \cancel{5}\}$$

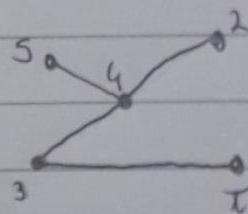
$$S = \{5, 4, 3, 2, \cancel{1}\}$$

$$S \rightarrow 2: \infty \cancel{9} \text{ (6)}$$

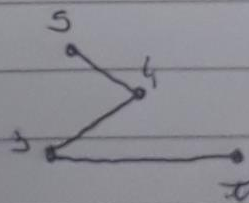
$$S \rightarrow 3: \infty \cancel{6} \text{ (5)}$$

$$S \rightarrow 4: \infty \text{ (4)}$$

$$S \rightarrow \cancel{1}: \infty \infty \cancel{15} \cancel{10} \text{ (10)}$$



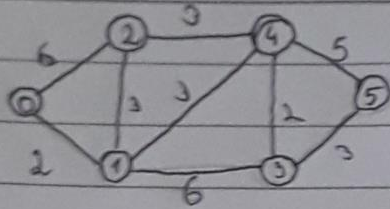
Memor caminho



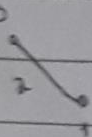
4)

Prüfung

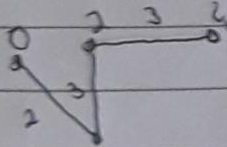
4. a)



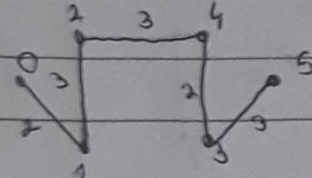
1:



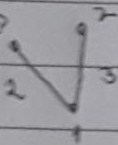
3:



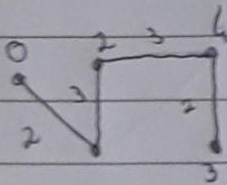
5:



2:



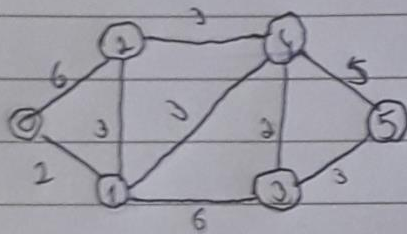
4:



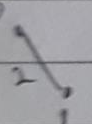
$$S = 2 + 3 + 3 + 2 + 3$$

$$S = 13$$

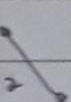
4. a) Kruskal



1:



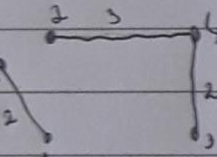
2:



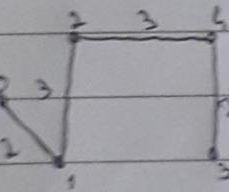
3:



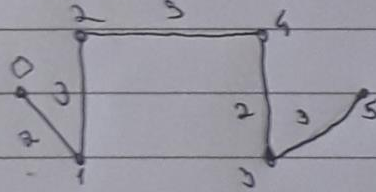
3:



4:

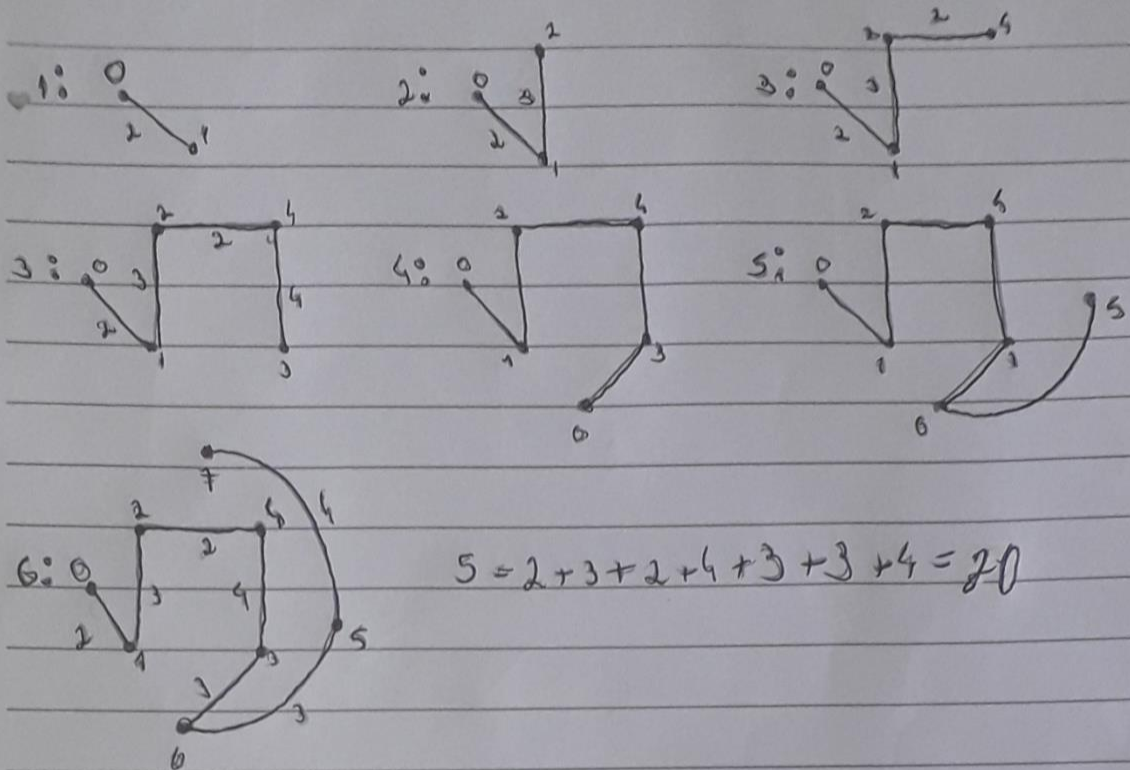


5:

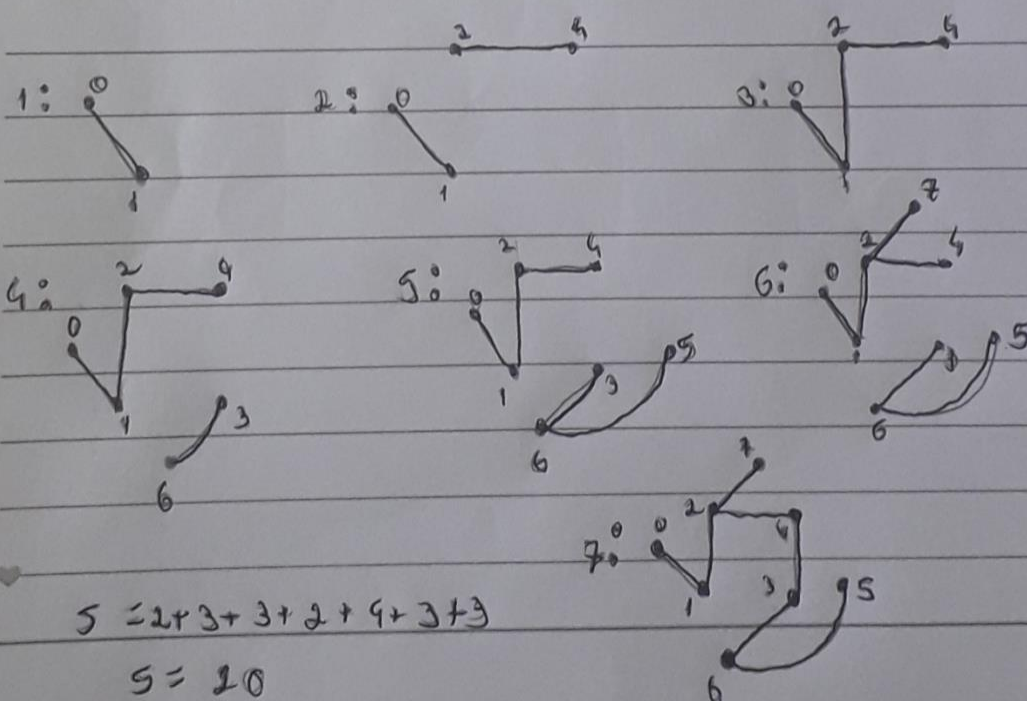


$$S = 2 + 3 + 3 + 2 + 3 = 13$$

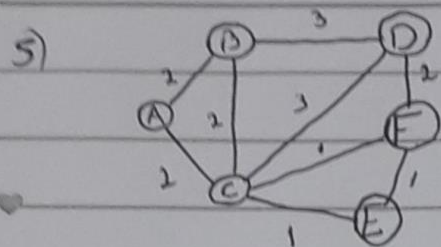
4 → b) Prim



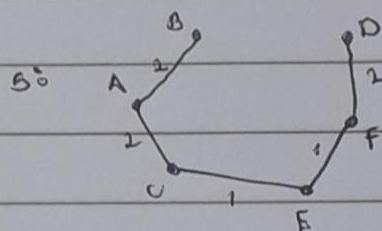
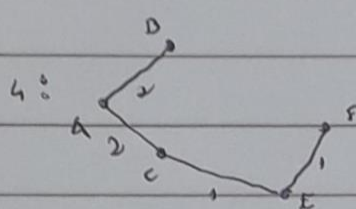
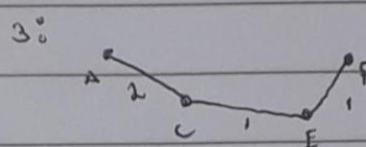
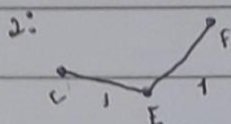
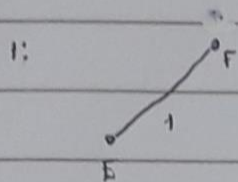
4 → b) Kruskal



5)



Por Kruskal:



6 vértices $\rightarrow (6-1) = 5$ arestas

custo = $1 + 1 + 2 + 2 + 2 = 8$ ✗