Conteúdos necessário para o pdf

Barbara Marques

1.Introdução:(especificar sobre o problema do labirinto)

Neste capítulo, será apresentado o problema do labirinto, definindo e especificando qual é o objetivo do trabalho.

Aqui você pode digitar um texto e adicionar uma imagem de um dos labirintos para facilitar a explicação do problema.

Exemplo de labirinto 10x10:
EXXXXXXXXX
000XXXXXXX
0X0000X000
0X0XX0X0X0
0X0XX000XS
0X0XXXXXXX
0X00000XXX
0XXXXX0XXX
000XX000XX
0XXXX0X0XX

Aqui está um exemplo de um dos labirintos para facilitar a explicação do problema.

2.Estruturas de dados:(Especificar todos as estruturas utilizadas em cada código)
Estruturas principais utilizado no primeiro código:

- **Fila first-in first-out**
- **Grafo**
- **Estrutura de nó para a implementação de uma lista encadeada usada para implementar uma fila(alocação dinâmica)**

Existem mais estruturas no primeiro código para ser analisado pelo gerenciador desse pdf

Estruturas principais utilizado no segundo código:

- **Pilha**
- **Grafo**
- **Estrutura de nó para a implementação de uma lista encadeada para implementar uma pilha(alocação dinâmica)**

3.Algoritmos utilizados e sua complexidade

Para esse Trabalho utilizamos algoritmos de Busca de Grafos

1. **Explicar o'que é Busca de Grafos**
2. **descrever os dois algoritmos utilizados em cada código sendo eles: Busca em largura(primeiro código) e Busca em profundidade(segundo código)**

Tem material do professor Iago sobre esses dois tipos de algoritmos para ajudar na explicação do pdf https://github.com/iagoac/dce529/blob/main/slides/aula_13.pdf
link acima é a aula sobre Busca de Grafos.

Diogo moreira e Gustavo marcelino - Conteúdos para os slides

1.Primeiro Slide

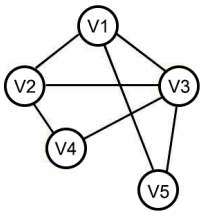
Ter os nomes de todos os participantes do trabalho com o nome completo e o tema(Busca de Grafos)
Abner Gomes Guimarães
Diogo…
Gustavo..
Marcelo..
Barbara…
Felipe…

Nome do Docente: Iago Augusto de Carvalho

2.Segundo Slide

Explicação sobre matriz de adjacência utilizando imagem para não ler muito texto durante a apresentação
Exemplo de imagem:

Matriz de Adjacência



	V1	V2	V3	V4	V5
V1	0	1	1	0	1
V2	1	0	1	1	0
V3	1	1	0	1	1
V4	0	1	1	0	0
V5	1	0	1	0	0

Imagem: Paulo Martins

O vértice V2 está conectado em v1,v3 e v4
Isso é representado na linha do vértice 2 com o número 1 em cada vértice correspondente

3.Terceiro Slide
Como representamos a matriz adjacente em código

Primeiro criamos uma matriz de vértices para representar os vértices existentes lendo o labirinto e transformando E, S e 0 como vértices em número 1(caminhos a serem percorridos)

```c
void verificacao_vertice(Grafo *grafo, FILE *arq) {
  char caractere;
  int linha = 0, coluna = 0;
  grafo->num_vertices = 0;
  Coordenada coordenada_inicio;
  Coordenada coordenada_fim;

  while ((caractere = fgetc(arq)) != EOF) {
    if (caractere == 'E' || caractere == 'S' || caractere == '0') {
      grafo->matriz_ver[linha][coluna] = 1;
      coluna++;
      if (caractere == 'E') {
        grafo->coordenada_inicio.linha = linha;
        grafo->coordenada_inicio.coluna = coluna;
      }
      if (caractere == 'S') {
        grafo->coordenada_fim.linha = linha;
        grafo->coordenada_fim.coluna = coluna;
      }
      grafo->num_vertices++;
    } else if (caractere == 'X') {
      grafo->matriz_ver[linha][coluna] = 0;
      coluna++;

    } else {
      linha++;
      coluna = 0;
    }
  }
}
```

E também pegamos as coordenadas do E e S para ser utilizado no método BFS e DFS

4.Quarto slide(pode ser colocado junto com o terceiro slide caso o gerenciador preferir)
Com as informações dos vértices existentes, precisamos criar uma matriz adjacente que vai ser utilizada para representar as conexões entre os vértices para representar o labirinto. Primeiro iniciamos a matriz de adjacência com nenhuma conexão(ou seja 0)

```c
void inicializar_matriz_adjacencia(Grafo *grafo, int num_vertices) {
  grafo->num_vertices = num_vertices;

  for (int i = 0; i < MAX_vet; i++) {
    for (int j = 0; j < MAX_vet; j++) {
      grafo->matriz_adj[i][j] = 0;
    }
  }
}
```

5.Quinto Slide
Construindo a matriz adjacência com as conexões utilizando a matriz de vértices

```c
void construir_matriz_adjacencia(int num_vertices, Coordenada *listaVertice) {

  int matriz_adj[num_vertices][num_vertices];
  for (int i = 0; i < num_vertices; i++) {
    for (int j = 0; j < num_vertices; j++) {

      if (&listaVertice[i] == &listaVertice[j]) {
        matriz_adj[i][j] = 0;
      } else {
        // verifica se conectado em cima
        if (listaVertice[j].linha - 1 == listaVertice[i].linha &&
            listaVertice[j].coluna == listaVertice[i].coluna) {
          matriz_adj[i][j] = 1;
        }
        // verifica se conectado em baixo
        else if (listaVertice[j].linha + 1 == listaVertice[i].linha &&
                 listaVertice[j].coluna == listaVertice[i].coluna) {
          matriz_adj[i][j] = 1;
        }
        // verifica se conectado na direita
        else if (listaVertice[j].coluna + 1 == listaVertice[i].coluna &&
                 listaVertice[j].linha == listaVertice[i].linha) {
          matriz_adj[i][j] = 1;

        }
        // verifica se conectado na esquerda
        else if (listaVertice[j].coluna - 1 == listaVertice[i].coluna &&
                 listaVertice[j].linha == listaVertice[i].linha) {
          matriz_adj[i][j] = 1;
        } else {
          matriz_adj[i][j] = 0;
        }
      }
```

Próximos slides:
Os próximos slides vai ser utilizado para explicar os códigos de busca, como por exemplo BFS(Busca em largura) e DFS(Busca de profundidade)
e apresentar as complexidades de cada algoritmo

Data provável de finalização dos códigos 30/03/2024

Avisos importantes: esse pdf é para ser utilizado apenas como guia para ajudar os gerenciadores da apresentação e criador do pdf original.

- **Pesquisar cada método de busca para estar informado sobre o projeto e utilizar nas apresentações.**
- **Utilizar os slides pré pronto mas mudar os códigos para que fique bom para cada imagem especifica(não apenas mudar a imagem do código LATEX)**
- **Não colocar muito texto**
- **Utilizar mais imagem para representar cada tópico**

- **Confirmar as estruturas de dados com cada programador do algoritmo**

Método BFS e DFS: Abner Gomes Guimarães, Felipe Correia e Marcelo Bernardino

Alguma reclamação ou sugestão de mudança? Vamos discutir isso com todos do grupo para chegarmos a uma solução que atenda a todos!

Agradeço e espero que esse guia ajude na criação dos slides e pdf.