

Principais Assuntos Implementados

1. Estruturas de Grafos:

Matriz de Adjacência:

Implementação de um grafo usando uma matriz quadrada para armazenar as arestas.

Vantagem: Acesso rápido à existência de arestas entre dois nós.

Desvantagem: Consumo de memória proporcional ao quadrado do número de nós.

Lista de Adjacência:

Implementação de um grafo usando um vetor de listas, onde cada nó armazena suas arestas.

Vantagem: Eficiente em memória para grafos esparsos.

Desvantagem: Acesso mais lento à existência de arestas específicas.

2. Operações Básicas:

Inserção de Nós:

Adição de novos nós ao grafo, com redimensionamento dinâmico da matriz ou lista.

Inserção de Arestas:

Adição de arestas entre nós, com suporte para grafos direcionados e não direcionados.

Em grafos ponderados, armazenamento do peso da aresta.

Exclusão de Nós:

Remoção de um nó e todas as suas arestas associadas.

Reindexação dos nós restantes para manter a consistência.

Exclusão de Arestas:

Remoção de uma aresta específica entre dois nós.

3. Propriedades do Grafo:

Grau Máximo:

Cálculo do maior grau entre todos os nós do grafo.

Em grafos direcionados, considere os graus de entrada e saída.

Completude:

Verifique se o grafo é completo (todos os nós estão conectados entre si).

Em grafos direcionados, verificação de todas as arestas possíveis.

Maior Menor Distância:

Cálculo da maior distância mínima entre todos os pares de nós alcançáveis.

Utilização do algoritmo de Dijkstra para encontrar as distâncias mínimas.

4. Algoritmos Implementados:

Dijkstra:

Algoritmo para encontrar o caminho mais curto entre dois nós em grafos ponderados.

Utilização de uma fila de prioridade para eficiência.

Reindexação após Exclusão:

Atualização dos IDs dos nós após a exclusão de um nó.

Garantia de que todas as referências às arestas sejam atualizadas corretamente.

5. Testes e Validação:

Grafos de Teste:

Criação de grafos direcionados, não direcionados, ponderados e não ponderados.

Exemplos de grafos completos, desconexos e esparsos.

Validação das Operações:

Testes de inserção e exclusão de nós e arestas.

Verificação das propriedades do grafo após cada operação.

Depuração:

Uso de logs para verificar o estado do grafo após cada operação.

Correção de problemas como reindexação incorreta e cálculo de distâncias.

6. Dificuldades e Soluções:

Reindexação após Exclusão:

Problema: Manter a consistência dos IDs dos nós após a exclusão.

Solução: Atualização manual de todas as referências às arestas.

Cálculo da Maior Menor Distância:

Problema: Distâncias infinitas entre nós não alcançáveis.

Solução: Ignorar pares de nós não alcançáveis no cálculo.

Eficiência:

Problema: Cálculos lentos em grafos grandes.

Solução: Uso de estruturas de dados eficientes, como filas de prioridade.

7. Conclusão:

Aprendizados:

Aplicação prática de conceitos teóricos de grafos.

Desenvolvimento de habilidades em programação e resolução de problemas.

Próximos Passos:

Implementação de algoritmos mais avançados, como Floyd-Warshall ou Bellman-Ford.

Extensão do projeto para suportar grafos com pesos negativos.