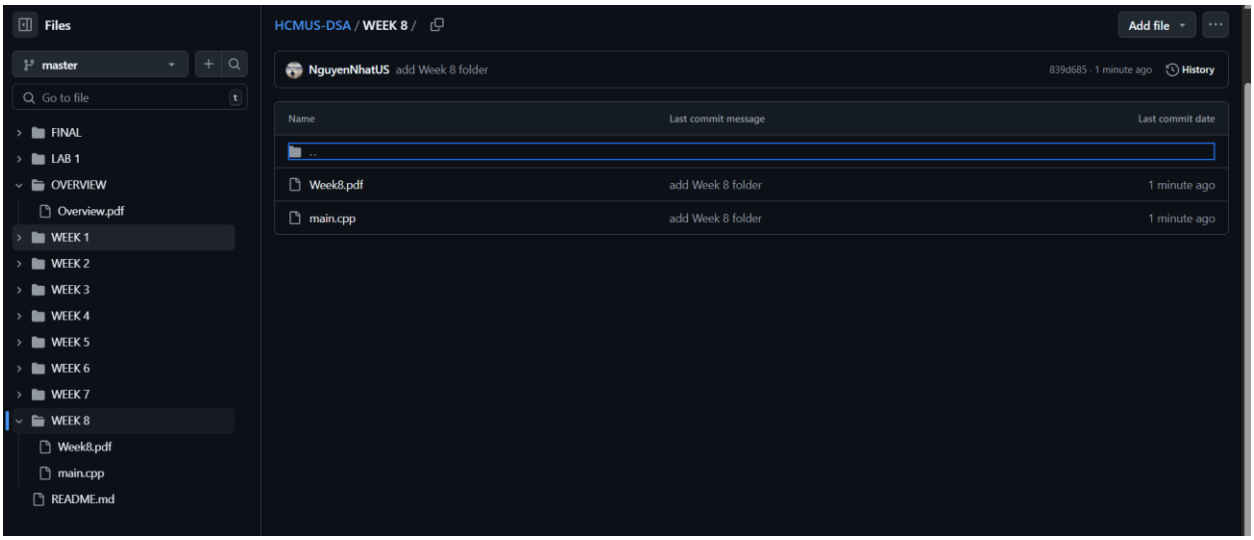
24120403

Nguyễn Lê Đức Nhật



1. convertMatrixToList(const string& filename)

Chức năng: Đọc ma trận kề từ file và chuyển thành danh sách kề.

Ý tưởng: Đọc từng dòng của file, nếu giá trị tại (i, j) = 1 thì thêm j vào danh sách kề của đỉnh i.

2. convertMatrixToList(const vector<vector<int>>& adjMatrix)

Chức năng: Chuyển ma trận kề sang danh sách kề.

Ý tưởng: Duyệt toàn bộ ma trận, nếu phần tử adjMatrix[i][j] khác 0 thì thêm j vào danh sách kề của i.

3. convertListToMatrix(const string& filename)

Chức năng: Đọc danh sách kề từ file và chuyển thành ma trận kề.

Ý tưởng: Mỗi dòng đầu chứa số đỉnh, sau đó mỗi dòng liệt kê các đỉnh kề với đỉnh tương ứng.

4. isDirected(const vector<vector<int>> & adjMatrix)

Chức năng: Kiểm tra xem đồ thị có hướng hay không.

Ý tưởng: Nếu tồn tại cặp (i,j) mà adj[i][j] != adj[j][i] thì là đồ thị có hướng.

5. countVertices & countEdges

Chức năng: Đếm số đỉnh và số cạnh.

Ý tưởng: Đếm kích thước ma trận và tổng các phần tử (chia 2 nếu vô hướng).

6. getIsolatedVertices

Chức năng: Tìm các đỉnh cô lập (không có cạnh nối nào).

Ý tưởng: Kiểm tra dòng và cột của từng đỉnh, nếu toàn 0 thì là cô lập.

7. isCompleteGraph

Chức năng: Kiểm tra đồ thị đầy đủ.

Ý tưởng: Số cạnh phải bằng n*(n-1)/2 trong đồ thị vô hướng.

8. isBipartite & isCompleteBipartite

Chức năng: Kiểm tra đồ thị có hai phần hay không (đầy đủ hoặc không)

Ý tưởng: Dùng tô màu bằng BFS. Với đồ thị đầy đủ hai phần, tổng cạnh phải bằng |A| * |B|.

9. convertToUndirectedGraph

Chức năng: Chuyển đồ thị có hướng thành vô hướng.

Ý tưởng: Nếu có cạnh i->j, thì tạo thêm j->i.

10. getComplementGraph

Chức năng: Tìm đồ thị bù.

Ý tưởng: Nếu (i,j) không có cạnh thì thêm cạnh vào kết quả.

11. findEulerCycle

Chức năng: Tìm chu trình Euler.

Ý tưởng: Dựa trên thuật toán Hierholzer. Kiểm tra điều kiện Euler rồi tìm chu trình.

12. dfsSpanningTree, bfsSpanningTree

Chức năng: Tạo cây khung bằng DFS hoặc BFS.

Ý tưởng: Tìm cây bao trùm bằng DFS hoặc BFS từ đỉnh gốc, tạo danh sách cạnh cây.

13. isConnected

Chức năng: Kiểm tra hai đỉnh có nối với nhau không.

Ý tưởng: Dùng DFS đánh dấu vùng liên thông, so sánh hai đỉnh.

14. dijkstra

Chức năng: Tìm đường đi ngắn nhất từ start đến end với trọng số không âm.

Ý tưởng: Dùng hàng đợi ưu tiên để cập nhật khoảng cách nhỏ nhất.

15. bellmanFord

Chức năng: Tìm đường đi ngắn nhất từ start đến end, có thể có trọng số âm.

Ý tưởng: Lặp cập nhật khoảng cách n-1 lần, kiểm tra chu trình âm.