**非连通图的遍历**

void DFS(GraphLnk&gl, int v, bool visited[]) {

cout << gl.NodeTable[v].data<<"-->";

visited[v] = true;

int w = GetFirstNeighbor(gl, gl.NodeTable[v].data);

while (w != -1) {

if (!visited[w]) {

DFS(gl, w, visited);

}

//一部分的深度结束后，向广度扩散

w = GetNextNeighbor(gl, gl.NodeTable[v].data, gl.NodeTable[w].data);

}

}

void Components(GraphLnk&gl) {

int n = gl.NumVertices;

bool \*visited = (bool\*)malloc(sizeof(bool)\*n);

assert(visited != nullptr);

for (int i = 0; i < n; i++) {

visited[i] = false;

}

for (int i = 0; i < n; i++) {

if (!visited[i]) {

DFS(gl, i, visited);

}

}

free(visited);

}