class FordFulkerson

{

static bool Bfs(int[,] rGraph, int Istok, int Stok, int[] parent, int V)

{

bool[] visited = new bool[V];

for (int i = 0; i < V; ++i) visited[i] = false;

Queue<int> queue = new Queue<int>();

queue.Enqueue(Istok);

visited[Istok] = true;

parent[Istok] = -1;

while (queue.Count != 0)

{

int u = queue.Dequeue();

for (int i = 0; i < V; i++)

{

if (visited[i] == false && rGraph[u, i] > 0)

{

if (i == Stok)

{

parent[i] = u;

return true;

}

queue.Enqueue(i);

parent[i] = u;

visited[i] = true;

}

}

}

return false;

}

int FordFulkersonAlgorithm(int[,] Graph, int Istok, int Stok, int V)

{

int u, v;

int[,] rGraph = new int[V, V];

for (u = 0; u < V; u++) for (v = 0; v < V; v++) rGraph[u, v] = Graph[u, v];

int[] parent = new int[V];

int maxFlow = 0;

while (Bfs(rGraph, Istok, Stok, parent, V))

{

int LocalFlow = int.MaxValue;

for (v = Stok; v != Istok; v = parent[v])

{

u = parent[v];

LocalFlow = Math.Min(LocalFlow, rGraph[u, v]);

}

for (v = Stok; v != Istok; v = parent[v])

{

u = parent[v];

rGraph[u, v] -= LocalFlow;

rGraph[v, u] += LocalFlow;

}

maxFlow += LocalFlow;

}

return maxFlow;

}

public static void Main()

{

int V = 5;

int[,] graph = new int[,] { {0, 20, 30, 10, 0},

{0, 0, 0, 40, 30},

{0, 0, 0, 10, 20},

{0, 0, 0, 0, 20},

{0, 0, 0, 0, 0} };

FordFulkerson fordFulkerson = new FordFulkerson();

Console.WriteLine("Максимальный поток: " + fordFulkerson.FordFulkersonAlgorithm(graph, 0, 4, V));

}

}print(f"Максимальный поток равен: {F}")