```
CreateRandomCommunityGraph[commCount_, commSize_] := (
  pNeig = 0.4;
  pNotNeig = 0.02;

  matrix = RandomChoice[{pNotNeig, 1 - pNotNeig} → {1, 0},
    {commSize * commCount, commSize * commCount}];

  Do[matrix[[commSize * (i - 1) + 1 ;; commSize * i,
     commSize * (i - 1) + 1 ;; commSize * i]] = RandomChoice[
     {pNeig, 1 - pNeig} → {1, 0}, {commSize, commSize}], {i, commCount}];
  Do[matrix[[i, i]] = 0, {i, commSize * commCount}];

  colors = {Red, Green, Blue, Black, White, Gray,
    Cyan, Magenta, Yellow, Brown, Orange, Pink, Purple, LightRed,
    LightGreen, LightBlue, LightGray, LightCyan , LightMagenta,
    LightYellow, LightBrown , LightOrange, LightPink, LightPurple};

  g = UndirectedGraph[AdjacencyGraph[matrix]];
  g
 )
```

```
Neighbours[g_, v_] :=
 DeleteCases[Flatten[Table[{e[[1]], e[[2]]}, {e, EdgeList[g, v ê _]}]], v]

CommonNeighbours[g_, v1_, v2_] :=
 Intersection[Neighbours[g, v1], Neighbours[g, v2]]

DropIds[list_, ids_] := list[[Complement[Table[i, {i, Length[list]}], ids]]]

DeleteRowCol[matrix_, ids_] :=
 matrix[[Complement[Table[i, {i, Length[matrix]}], ids],
  Complement[Table[i, {i, Length[matrix]}], ids]]]

MeanTable[t1_, n1_, t2_, n2_] :=
 Table[(t1[[i]] * n1 + t2[[i]] * n2) / (n1 + n2), {i, Length[t1]}]

OneStep[args_] := (
  matrix = args[[1]];
  vertices = args[[2]];
  p = Position[matrix, Max[matrix]][[1]];
  c1 = vertices[[p[[1]]]];
  c2 = vertices[[p[[2]]]];
  n1 = Length[c1];
  n2 = Length[c2];

  p1 = DropIds[matrix[[p[[1]]]], p];
  p2 = DropIds[matrix[[p[[2]]]], p];
  pmean = MeanTable[p1, n1, p2, n2];
  m = DeleteRowCol[matrix, p];
  m = Table[AppendTo[m[[i]], pmean[[i]]], {i, Length[pmean]}];
  m = AppendTo[m, AppendTo[pmean, 0]];
  v = Append[DeleteCases[vertices, c1 | c2], Flatten[{c1, c2}]];
  {m, v}
 )
GeneralClusterGraph[g_] := (
  matrix = Table[If[v1 == v2, 0, Length[CommonNeighbours[g, v1, v2]]],
    {v1, VertexList[g]}, {v2, VertexList[g]}];
  vertices = Table[{v}, {v, VertexList[g]}];
  size = VertexCount[g];
  result = NestList[OneStep, {matrix, vertices}, size - 2];
  result
 )
ClusterGraph[g_, nClusters_] := (
  result = GeneralClusterGraph[g];
  size = VertexCount[g];
  nIterations = size - nClusters + 1;
  result = result[[nIterations]];
  result[[2]]
 )
```

```
ClusterColorGraph[g_, clusters_] := (
  newColors = Flatten[Table[clusters[[nc, i]] → colors[[nc]],
     {nc, Length[clusters]}, {i, Length[clusters[[nc]]]}]];
  Graph[EdgeList[g], VertexStyle → newColors, ImageSize → 600]
 )
```

```
InnerClusterDensity[graph_, verticesList_] := (
  connections = Table[If[MemberQ[verticesList, e[[1]]],
     If[MemberQ[verticesList, e[[2]]], {1, 1}, {0, 1}], If[
      MemberQ[verticesList, e[[2]]], {0, 1}, {0, 0}]], {e, EdgeList[graph]}];
  (*inner=If[Length[verticesList]==1,1,Total[connections[[All,1]]]/
      (Length[verticesList]*(Length[verticesList]-1))];*)
  (*outer=(Total[connections[[All,2]]]-Total[connections[[All,1]]])/
    Length[verticesList];
  density=inner/outer;
  density*)
  Total[connections[[All, 1]]] / Total[connections[[All, 2]]] /
   (Length[verticesList])
 )
MeanClustersDensity[graph_, clusters_] := (
  densitySum = Total[Table[
     InnerClusterDensity[graph, verticesList], {verticesList, clusters}]];
  meanDensity = densitySum / Length[clusters];
  N[meanDensity]
 )
```

```
ClusterItPlease[graph_] := (
  results = GeneralClusterGraph[graph];
  densities =
   Table[MeanClustersDensity[graph, result[[2]]], {result, results}];
  clusters = results[[Position[densities, Max[densities]][[1]]]][[1, 2]];
  clusters
 )
```

```
commSize = 20;
commCount = 4;
g = CreateRandomCommunityGraph[commCount, commSize];
nClusters = 4;
{ClusterColorGraph[g, ClusterGraph[g, nClusters]],
 ClusterColorGraph[g, ClusterItPlease[g]]}
```

```
g = ExampleData[{"NetworkGraph", "AmericanCollegeFootball"}];
nClusters = 4;
{ClusterColorGraph[g, ClusterGraph[g, nClusters]],
 ClusterColorGraph[g, ClusterItPlease[g]]}
```

```
SetDirectory[
  "C:\\Users\\Ania\\Desktop\\SIECI\\projekt10\\Label_propagation_community
    _detection\\Graph_examples"];
```

```
GetClustersFromFile[file_] := (
  clustersImported = Import[file, "Table", "FieldSeparators" → ","];
  clusters = Drop[Map[#1[[1]] ê #1[[2]] &, clustersImported], 1];
  clusters = Table[
    Table[v[[1]], {v, Cases[clusters, _ ê i]}], {i, 0, clusters[[-1, 2]]}];
  clusters
 )
```

```
CompareGraphsWithout[graphFile_, clustersFile_] := (
  graphImported = Import[graphFile, "Table", "FieldSeparators" → ","];
  edges = Map[#1[[1]] ê #1[[2]] &, graphImported];
  edges =
   Table[If[e[[1]] < e[[2]], e[[1]] ê e[[2]], e[[2]] ê e[[1]]], {e, edges}];
  edges = DeleteDuplicates[edges];
  g = Graph[edges];
  g1 = ClusterColorGraph[g, ClusterItPlease[g]];
  g2 = ClusterColorGraph[g, GetClustersFromFile[clustersFile]];
  {g1, g2}
 )
CompareGraphsWith[graphFile_, clustersFile_] := (
  graphImported = Import[graphFile, "Table", "FieldSeparators" → ","];
  edges = Map[#1[[1]] ê #1[[2]] &, graphImported];
  edges =
   Table[If[e[[1]] < e[[2]], e[[1]] ê e[[2]], e[[2]] ê e[[1]]], {e, edges}];
  edges = DeleteDuplicates[edges];
  g = Graph[edges];
  clusters = GetClustersFromFile[clustersFile];
  g1 = ClusterColorGraph[g, ClusterGraph[g, Length[clusters]]];
  g2 = ClusterColorGraph[g, clusters];
  {g1, g2}
 )
```

```
CompareGraphsWithout["example.csv", "example_communities.csv"]
```

```
CompareGraphsWith["example.csv", "example_communities.csv"]
```

```
CompareGraphsWith["example3.csv", "example3_communities.csv"]
```

```
CompareGraphsWithout["example3.csv", "example3_communities.csv"]
```

```
CompareGraphsWith["example4.csv", "example4_communities.csv"]
```

```
CompareGraphsWithout["example4.csv", "example4_communities.csv"]
```

```
CompareGraphsWith["Friendship-network_data_2013.csv",
 "Friendship-network_data_2013_communities.csv"]
```

```
CompareGraphsWithout["Friendship-network_data_2013.csv",
 "Friendship-network_data_2013_communities.csv"]
```