

Мухамадиев Владимир

Задание 9

Загрузка и предварительная обработка

```
In[1]:= (*Get["https://raw.githubusercontent.com/szhorvat/IGraphM/master/IGInstaller.m"]*)  
      |взять  
  
In[2]:= Needs["IGraphM`"]  
      |необходимо  
IGraph/M 0.4 (April 2, 2020)  
Out[2]:= Evaluate IGDokumentation[] to get started.  
  
In[3]:= karate = ExampleData[{"NetworkGraph", "ZacharyKarateClub"}];  
      |данные для примеров  
  
In[4]:= pc = {{1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 17, 18, 20, 22},  
      {10, 15, 16, 19, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34}};
```

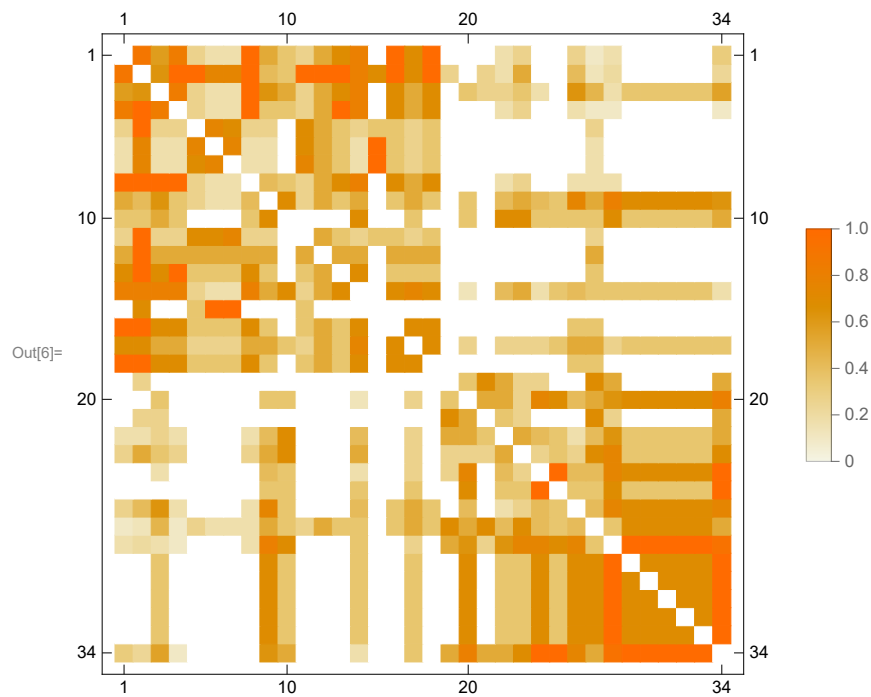
1. Similarity matrix

```
In[5]:= GraphSimilarityMatrix[graph_] :=  
  Module[{gdm = GraphDistanceMatrix[graph], jm = {}, am = AdjacencyMatrix[graph],  
    |программный ... |матрица расстояний на графе |матрица смежности  
    dl = VertexDegree[graph], vc = VertexCount[graph]}, jm = Table[  
    |степень вершины |число вершин |таблица значений  
    If[i == j, 0, Length[Position[{gdm[[i]], gdm[[j]]}^T, {1, 1}]]], {i, 1, vc}, {j, 1, vc}];  
    |условный оп... |длина |позиция по образцу  
    Table[(jm[[i, j]] + am[[i, j]]) / (Min[dl[[i]], dl[[j]]] + 1 - am[[i, j]]), {i, 1, vc}, {j, 1, vc}]]  
    |таблица значений |минимум
```

Матрица схожести заданного графа

```
In[6]:= MatrixPlot[GraphSimilarityMatrix[karate], PlotLegends -> Automatic]
```

[визуализация матрицы] [легенды графика] [автоматически]



2. Ravasz Algorithm

```

In[7]:= Ravasz[graph_] :=
Module[{gsm = GraphSimilarityMatrix[graph], clusters = Partition[VertexList[graph], 1],
  программный модуль разбиение... список вершин графа
  ce = {Partition[VertexList[graph], 1]}, max = {}, temp = {}},
  разбиение... список вершин графа
  Do[max = Sort[RandomChoice[Position[gsm, Max[gsm]]]];
  оператор сор... случайный вы... позиция по обр... максимум
  clusters[[max[[1]]] = Flatten[{clusters[[max[[1]]], clusters[[max[[2]]]]];
  уплостить
  clusters = Delete[clusters, max[[2]]];
  удалить элемент
  AppendTo[ce, clusters];
  добавить в конец к
  temp = (gsm[[max[[1]]] + gsm[[max[[2]]]]) / Length[clusters[[max[[1]]]]];
  длина

  temp[[max[[1]]] = 0;
  gsm[[max[[1]]] = temp;
  gsm = gsmT;
  gsm[[max[[1]]] = temp;
  gsm = Delete[gsm, max[[2]]];
  удалить элемент

  gsm = gsmT;
  gsm = Delete[gsm, max[[2]], {1, 1, VertexCount[graph] - 1}];
  удалить элемент число вершин
  ce]

```

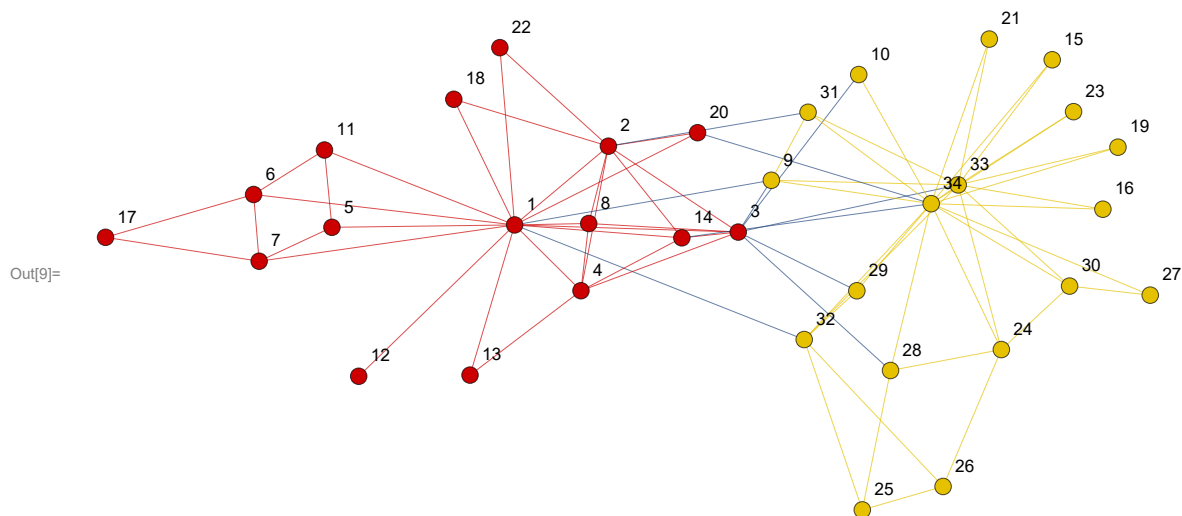
Разбиение на сообщества по алгоритму

```

In[8]:= ac = Ravasz[karate][[-2]];

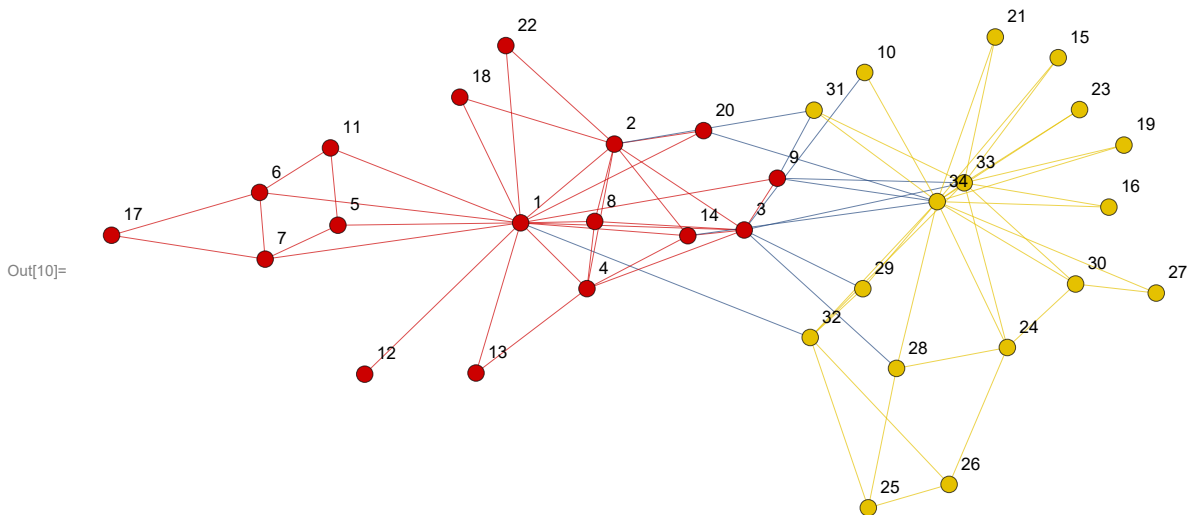
In[9]:= HighlightGraph[karate, Subgraph[karate, #] & /@ ac,
  граф с подкраской подграф
  VertexLabels -> "Name", ImageSize -> Large]
  метки для вершин размер изоб... крупный

```



Эталонное разбиение на сообщества

```
In[10]:= HighlightGraph[karate, Subgraph[karate, #] & /@ pc,
  _[граф с подкраской]      _[подграф]
  VertexLabels -> "Name", ImageSize -> Large]
  _[метки для вершин]      _[размер изоб...] _[крупный]
```

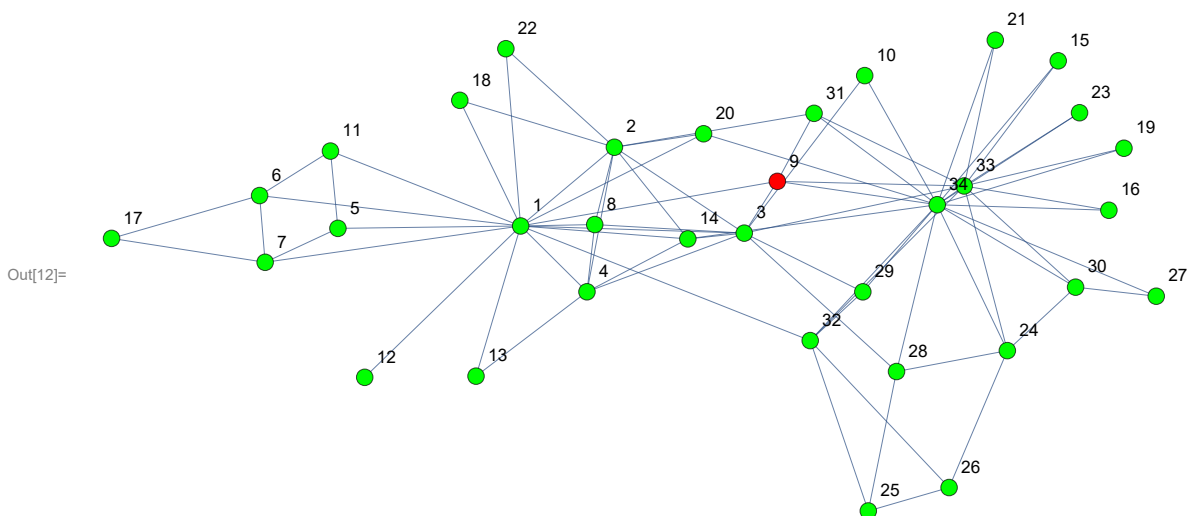


Вершины, которые оказались в других сообществах по сравнению с эталоном

```
In[11]:= nt = Flatten[{Intersection[ac[[1]], pc[[2]]], Intersection[ac[[2]], pc[[1]]]}]
  _[уплостить] _[пересечение]      _[пересечение]
```

Out[11]= { 9 }

```
In[12]:= Graph[karate, VertexStyle ->
  _[граф]      _[стиль вершины]
  Flatten[Append[Map[# -> Green &, Complement[ac[[1]] U ac[[2]], nt]], Map[# -> Red &, nt]],
  _[уплостить] _[добав...] _[преобр...] _[зелёный] _[дополнение]      _[преобр...] _[красный]
  VertexLabels -> "Name", ImageSize -> Large]
  _[метки для вершин]      _[размер изоб...] _[крупный]
```



3. Центральность ребер в сети

```

In[13]:= EdgeRandomWalkCentrality[graph_, n_] :=
Module[{v1 = VertexList[graph], e1 = EdgeList[graph],
  |программный... |список вершин графа |список рёбер
  temp = {}, k = 0, tempdata = {}, out = ConstantArray[{}, n]}, Do[k = 0;
  |постоянный массив |оператор цикла
  tempdata = ConstantArray[{}, VertexCount[graph]^2];
  |постоянный массив
  Do[k++;
  |оператор цикла
  If[i == j, tempdata[[k]] = {},
  |условный оператор
  temp = IGRandomEdgeWalk[graph, v1[[i]], 10 EdgeCount[graph]];
  |число рёбер
  While[FirstPosition[temp, v1[[j]]] == Missing["NotFound"],
  |цикл... |позиция первого по образцу |пропуск
  temp = IGRandomEdgeWalk[graph, v1[[i]], 10 EdgeCount[graph]]];
  |число рёбер
  tempdata[[k]] = temp[[1 ;; FirstPosition[temp, v1[[j]]][[1]]],
  |позиция первого по образцу
  {i, 1, VertexCount[graph]}, {j, 1, VertexCount[graph]}];
  |число вершин |число вершин
  tempdata = DeleteCases[tempdata, {}];
  |удалить случаи по образцу
  tempdata = Flatten[Table[Normal[Counts[tempdata[[i]]], {i, 1, Length[tempdata]}]];
  |уплостить |табл... |норма... |встречаемость элементов |длина
  tempdata =
  Counts[DeleteCases[{tempdata[[All, 1]], tempdata[[All, 2]]}^T, {_, _?EvenQ}][[All, 1]];
  |встре... |удалить случаи по образцу |всё |всё |чётное... |всё
  tempdata = Table[tempdata[e1[[i]]], {i, 1, Length[e1]}];
  |таблица значений |длина
  out[[1]] = tempdata, {1, 1, n}];
N[Mean[out]]
|... |среднее значение

```

Для заданного графа

```

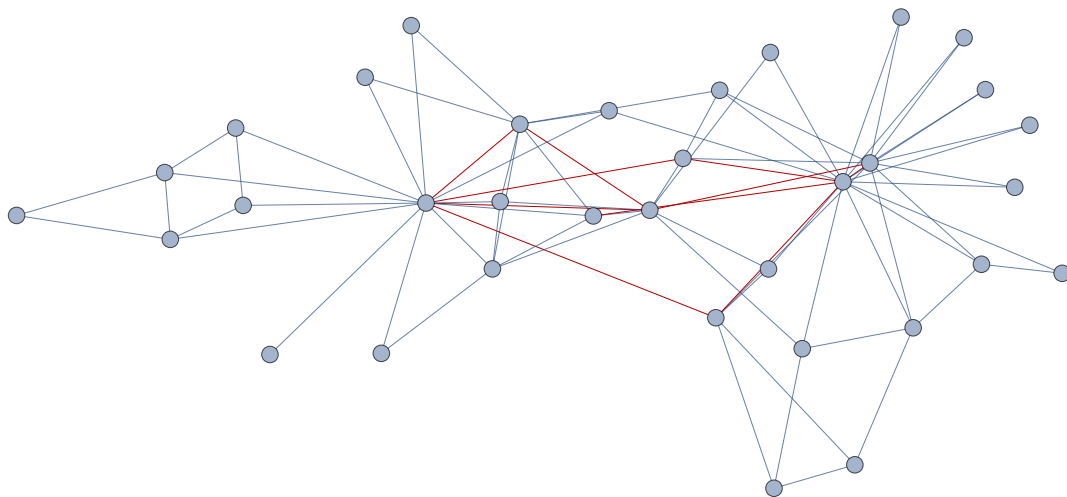
In[14]:= erwc = EdgeRandomWalkCentrality[karate, 100];

```

Визуализация топ 10

```
In[15]:= Graph[karate, GraphHighlight →
  [граф [выделить в графе
    ReverseSortBy[{EdgeList[karate], erwc}^T, Last][[1 ;; 10, 1]], ImageSize → Large]
  [сортировка в обр... [список рёбер [последний [размер изоб... [крупный
```

Out[15]=



Сравнение с топ 10 центральности по посредничеству

```

In[16]:= Grid[
  |таблица
  Prepend[Prepend[Flatten[{ReverseSortBy[{EdgeList[karate], erwc}]^T, Last][1 ;; 10]]^T,
    |добавит...|добавит...|уплостить |сортировка в обр... |список рёбер |последний
    ReverseSortBy[{EdgeList[karate], EdgeBetweennessCentrality[karate]}^T, Last][
      |сортировка в обр... |список рёбер |центральность рёбер по посредничеству |последн
      1 ;; 10]]^T, 1]^T, {"Ребра", "Значения", "Ребра", "Значения"}],
    {"Топ 10 центральности\нпо случайному блужданию", SpanFromLeft,
      |соединить с левым
      "Топ 10 центральности\нпо посредничеству", SpanFromLeft}], Frame → All,
      |соединить с левым |рамка |всё
  Background → {None, {{{Pink, Lighter[Blue, 0.7]}}, {1 → Gray, 2 → Gray}}},
    |фон |ни одног... |роз... |более с... |синий |серый |серый
  Spacings → {1, 1}]
  |размер зазора

```

Топ 10 центральности по случайному блужданию		Топ 10 центральности по посредничеству	
Ребра	Значения	Ребра	Значения
3 ↔ 1	303.92	32 ↔ 1	142.786
32 ↔ 1	302.69	7 ↔ 1	87.6667
33 ↔ 3	302.32	6 ↔ 1	87.6667
9 ↔ 1	298.75	3 ↔ 1	87.2778
2 ↔ 1	297.18	9 ↔ 1	83.2968
34 ↔ 14	295.97	33 ↔ 3	77.4032
34 ↔ 33	294.39	34 ↔ 14	76.0984
34 ↔ 9	293.04	34 ↔ 20	66.627
3 ↔ 2	292.85	12 ↔ 1	66.
34 ↔ 32	291.46	34 ↔ 27	60.9143

Out[16]=