

Мухамадиев Владимир

Задание 1

Загрузка и предварительная обработка

```
In[1]:= data = Import[NotebookDirectory[] <> "\\bio-diseasome.txt", "Data"];  
          | импорт | директория файла блокнота  
  
In[2]:= e1 = Table[data[[i, 1]] → data[[i, 2]], {i, 1, Length[data]}];  
          | таблица значений | длина  
  
In[3]:= g = Graph[e1];  
          | граф  
  
In[4]:= l = {{113, 25}, {389, 20}, {457, 26}, {329, 26}, {252, 26}, {66, 25}, {110, 25}};
```

1. Чему равно число вершин, число ребер в сети?

```
In[5]:= VertexCount[g]  
          | число вершин
```

Out[5]= 516

```
In[6]:= EdgeCount[g]  
          | число рёбер
```

Out[6]= 1188

2. Является ли сеть направленной?

```
In[7]:= DirectedGraphQ[g]  
          | ориентированный граф?
```

Out[7]= False

3. Чему равна средняя степень вершины?

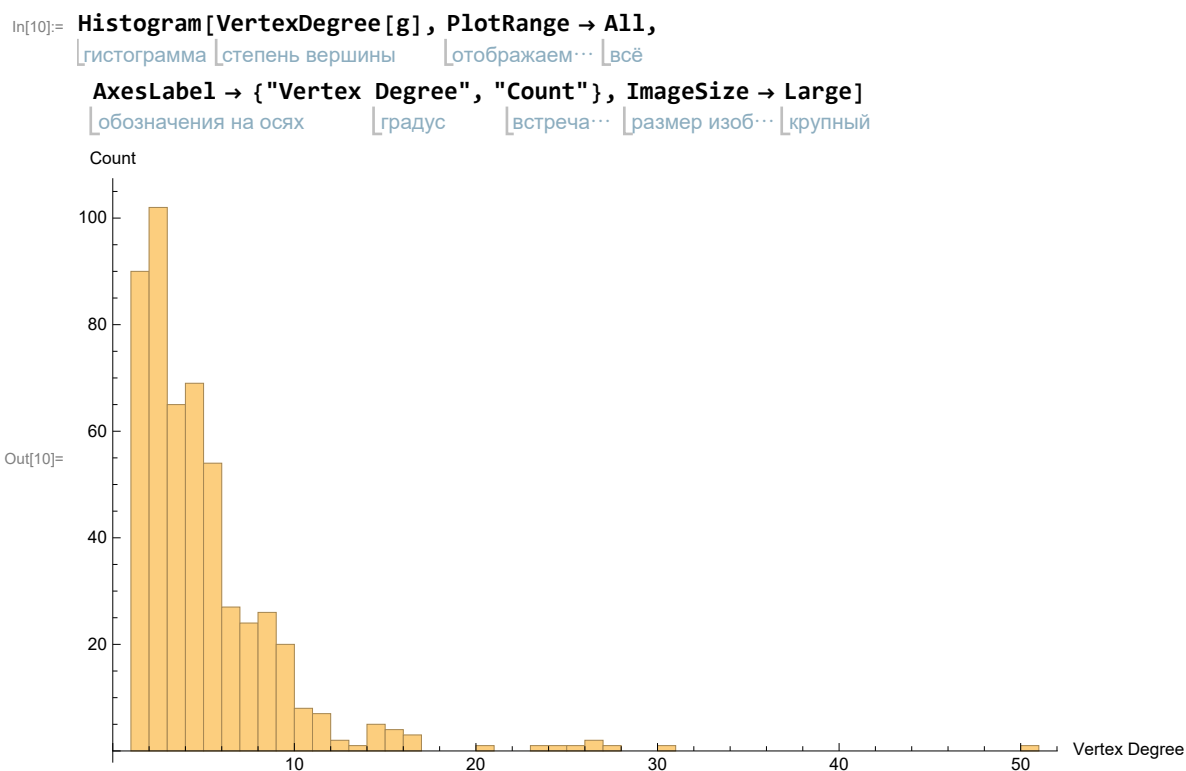
```
In[8]:= N[Mean[VertexDegree[g]]]  
          | ... | сре... | степень вершины
```

Out[8]= 4.60465

4. Перечислите узлы с максимальным значением степени.

```
In[9]:= Module[{in1 = ReverseSortBy[{VertexList[g], VertexDegree[g]}^T, Last],
  [программный ... [сортировка в обр... [список вершин г... [степень вершины [последний
    in2 = Max[VertexDegree[g]], i = 0, out = {}], While[in1[[i + 1, 2]] == in2, i++];
    [ма... [степень вершины [цикл-пока
    out = in1[[1 ;; i, 1]];
    out]
Out[9]= {93}
```

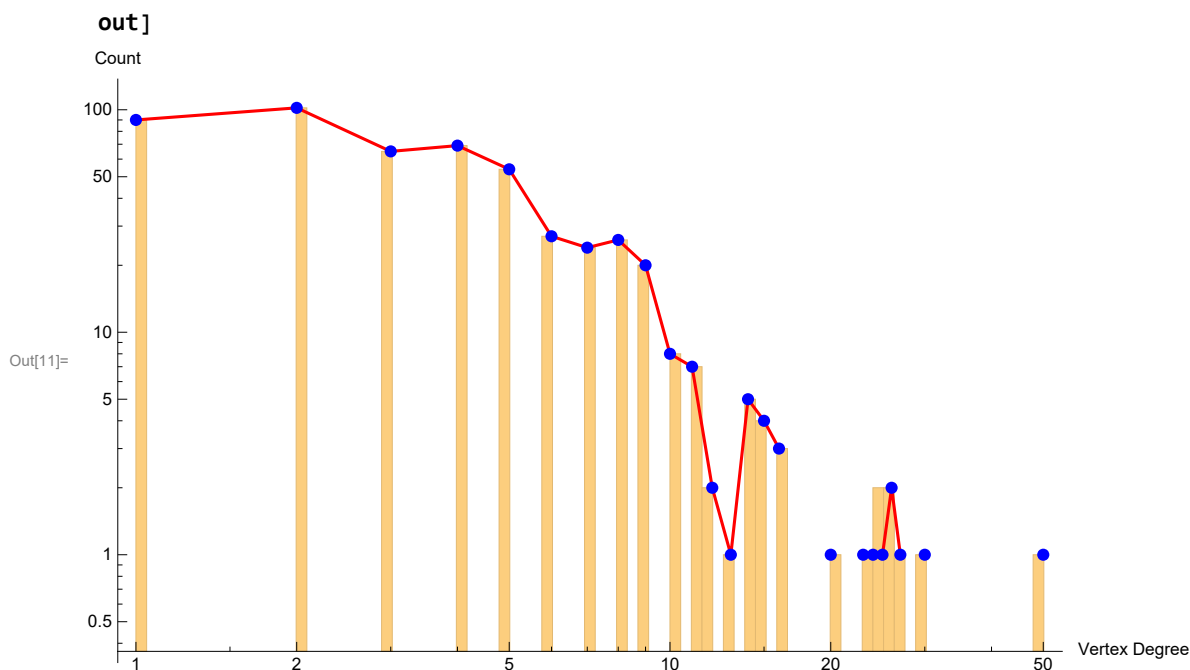
5. Постройте распределение по степеням связности и огибающую распределения в двойном логарифмическом масштабе.



```

In[11]:= Module[
  программный модуль
  {in1 = Histogram[VertexDegree[g], {"Log", Max[VertexDegree[g]]}, {"Log", "Count"},
    гистограмма степень вершины нат... ма... степень вершины натур... встречаемост
    PlotRange → All, AxesLabel → {"Vertex Degree", "Count"}, ImageSize → Large],
    отображаем... всё обозначения на осях градус встреча... размер изоб... крупный
    in2 = {Table[i, {i, 1, Max[VertexDegree[g]]}], HistogramList[VertexDegree[g]] [[2]]^T,
    таблица значений ма... степень вершины гистограммные ... степень вершины
    in3 = {}, out = {}, in3 = ListLogLogPlot[{in2, in2}, Joined → {True, False},
    диаграмма разброса данных в ло... соединён... ист... ложь
    PlotStyle → {Red, Blue}, PlotRange → All, ImageSize → Large];
    стиль графика кр... синий отображаем... всё размер изоб... крупный
    out = Show[in1, in3];
    показать
  ]

```



6. Сколько вершин имеют степень больше 20?

```

In[12]:= Module[{in = VertexDegree[g], out = 0}, Do[If[in[[i]] > 20, out++], {i, 0, Length[in]}];
  программный... степень вершины ... условный оператор длина
  out]

```

Out[12]= 8

7. Каких пар (индекс вершины, степень вершины) нет в вашем списке?

```

In[13]:= Module[{in1 = SortBy[{VertexList[g], VertexDegree[g]}^T, First], in2 = SortBy[1, First],
  программный ... сортиро... список вершин г... степень вершины первый сортирова... первый
  out = {}, Do[If[in2[[i]] ≠ in1[[in2[[i], 1]]], AppendTo[out, in2[[i]]], {i, 1, Length[in2]}];
  ... условный оператор добавить в конец к длина
  out]

```

Out[13]= {{66, 25}, {110, 25}, {329, 26}}

8. Сколько всего треугольников в сети?

```
In[14]:= 
$$\frac{\text{Total}[\text{Diagonal}[\text{MatrixPower}[\text{Normal}[\text{AdjacencyMatrix}[g]], 3]]}{6}$$

Out[14]= 1360
```

9. Чему равен средний коэффициент кластеризации в сети?

```
In[15]:= N[Mean[LocalClusteringCoefficient[g]]]
Out[15]= 0.63583
```

10. А чему равен коэффициент транзитивности сети?

```
In[16]:= N[GlobalClusteringCoefficient[g]]
Out[16]= 0.430471
```

11. Сколько процентов узлов из имеющих максимальную кластеризацию имеют степень два (т.е. являются вершинами треугольников)?

```
In[17]:= Module[{in = {LocalClusteringCoefficient[g], VertexDegree[g]}^T, n = 0, out = 0},
  Do[If[in[[i, 1]] == 1, n++], {i, 1, Length[in]}];
  If[in[[i, 2]] == 2, out++], {i, 1, Length[in]}];
  out = Quantity[N[ $\frac{\text{out}}{n} 100$ ], "Percent"];
  out]
Out[17]= 36.0902%
```

12. Какова максимальная степень вершины с минимальной кластеризацией?

```
In[18]:= Module[{in = {LocalClusteringCoefficient[g], VertexDegree[g]}^T, out = 0},
  Do[If[in[[i, 1]] == 0 && in[[i, 2]] > out, out = in[[i, 2]], {i, 1, Length[in]}];
  out]
Out[18]= 5
```

13. Какова кластеризация хаба (вершины с максимальной

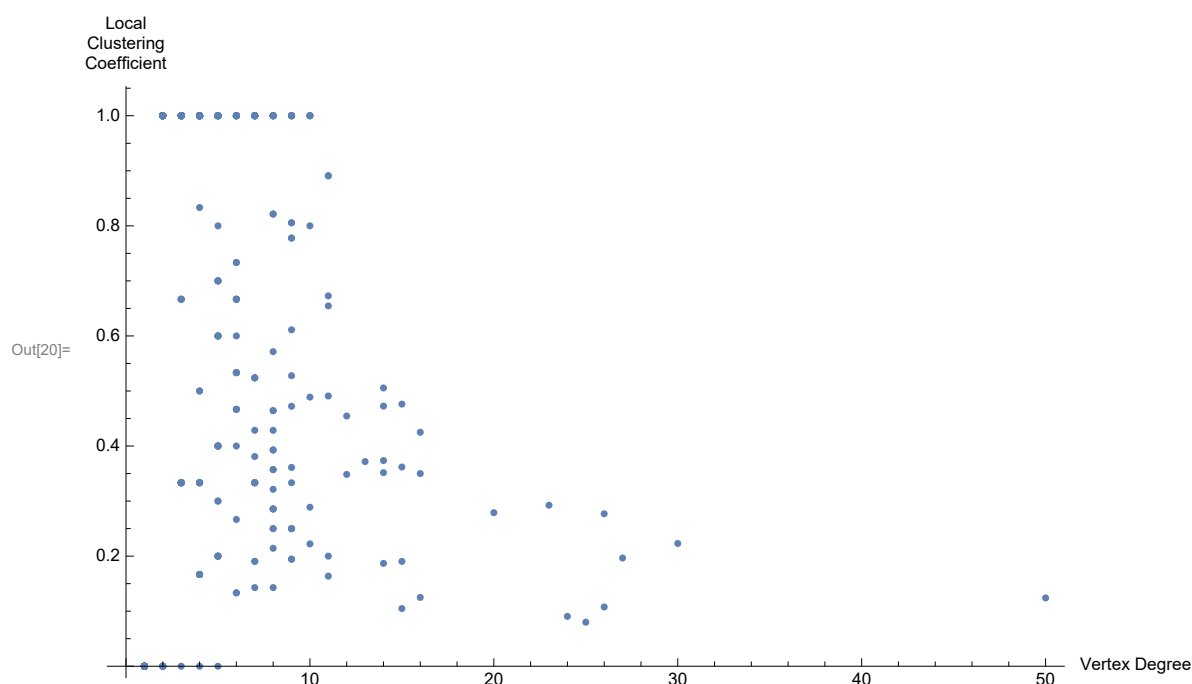
степенью)?

```
In[19]:= N[ReverseSortBy[{VertexDegree[g], LocalClusteringCoefficient[g]}^T, First][[1, 2]]
```

```
Out[19]= 0.124082
```

14. Постройте диаграмму рассеяния, по оси X - значение степени узла, по оси Y - его коэффициент кластеризации.

```
In[20]:= ListPlot[{VertexDegree[g], LocalClusteringCoefficient[g]}^T, PlotRange -> All,
  AxesLabel -> {"Vertex Degree", "Local\nClustering\nCoefficient"}, ImageSize -> Large]
```



15. Верно ли, что все узлы с кластеризацией, превышающей среднее значение имеют степень меньше 16?

```
In[21]:= Module[{in1 = {VertexDegree[g], LocalClusteringCoefficient[g]}^T,
  in2 = Mean[LocalClusteringCoefficient[g]], out = True},
  Do[If[in1[[i, 1]] > 16 && in1[[i, 2]] > in2, out = False], {i, 1, Length[in1]}];
  out]
```

```
Out[21]= True
```

16. Чему равен средний кратчайший путь в сети?

```
In[22]:= N[  
  Total[Flatten[GraphDistanceMatrix[g]]]  
  VertexCount[g]^2 - VertexCount[g]
```

Out[22]= 6.50899

17. А чему равен средний путь от хаба до всех остальных вершин?

```
In[23]:= N[  
  Mean[GraphDistance[g, ReverseSortBy[{VertexList[g], VertexDegree[g]}^T, Last][1, 1]]]
```

Out[23]= 4.38178

18. Чему равен диаметр сети?

```
In[24]:= GraphDiameter[g]
```

Out[24]= 15

19. Между какой парой вершин (i,j) число путей длины 4 наибольшее?

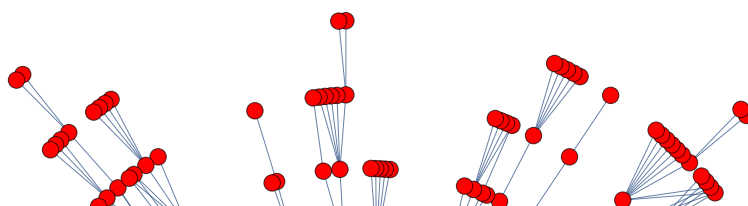
```
In[25]:= Module[{in1 = VertexList[g], in2 = VertexCount[g],  
  in3 = MatrixPower[Normal[AdjacencyMatrix[g]], 4], max = 0, out = {0, 0}},  
  Do[If[in3[[i, j]] > max && i ≠ j, max = in3[[i, j]],  
    out = {in1[[i]], in1[[j]]}, {i, 1, in2}, {j, 1, in2}];  
  out]
```

Out[25]= {71, 93}

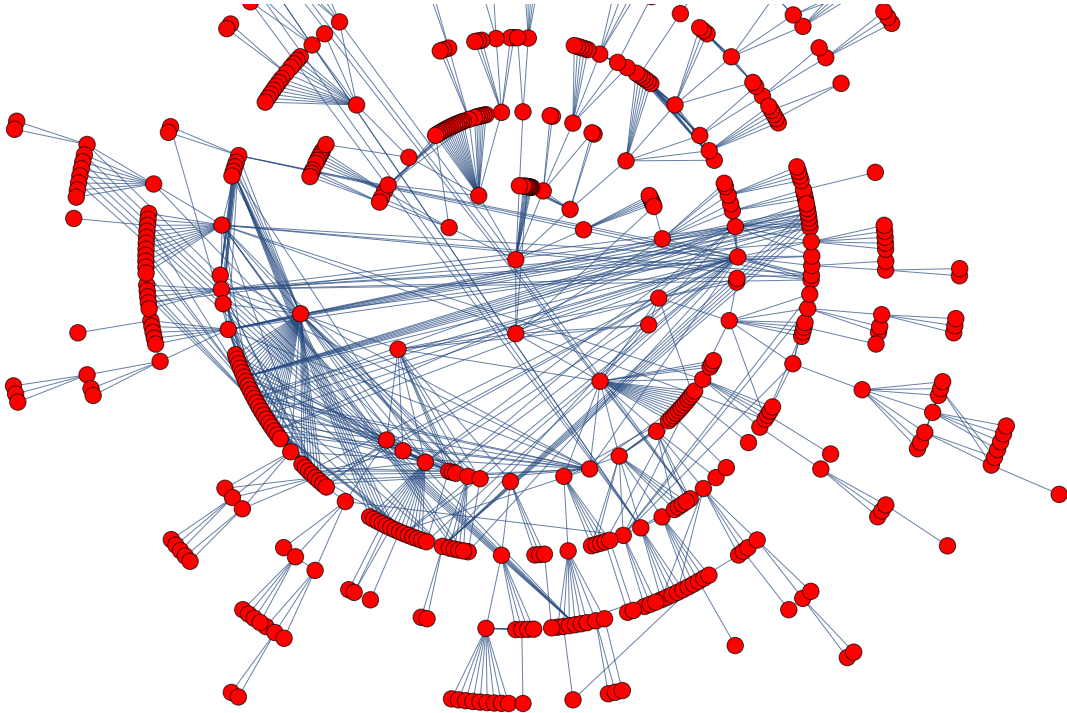
20. Визуализация сети.

```
In[26]:= Table[Graph[e1, GraphLayout → 1, PlotLabel → 1, VertexStyle → Red, ImageSize → Large],  
  {1, {"RadialEmbedding", "BalloonEmbedding", "SpringElectricalEmbedding"}}]
```

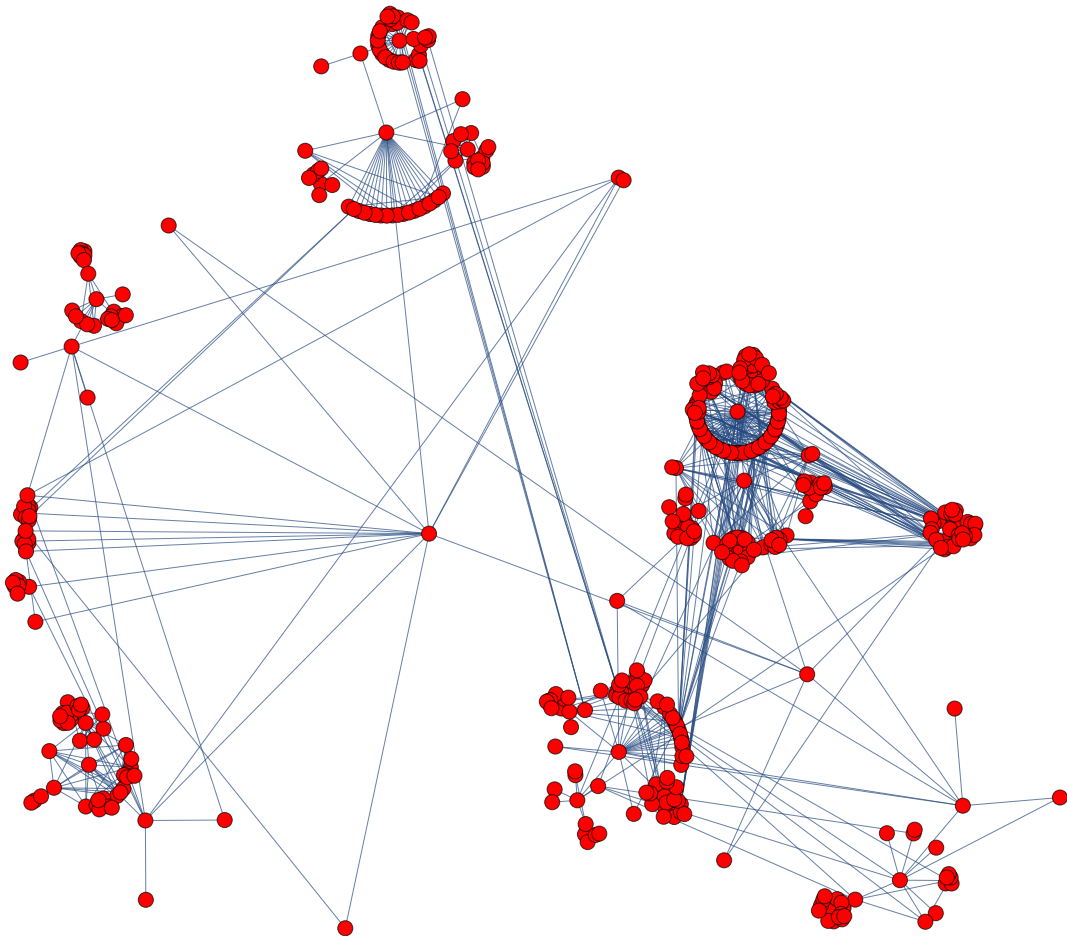
RadialEmbedding



Out[26]= {



BalloonEmbedding



SpringElectricalEmbedding

