

Мухамадиев Владимир

Задание 3

Загрузка и предварительная обработка

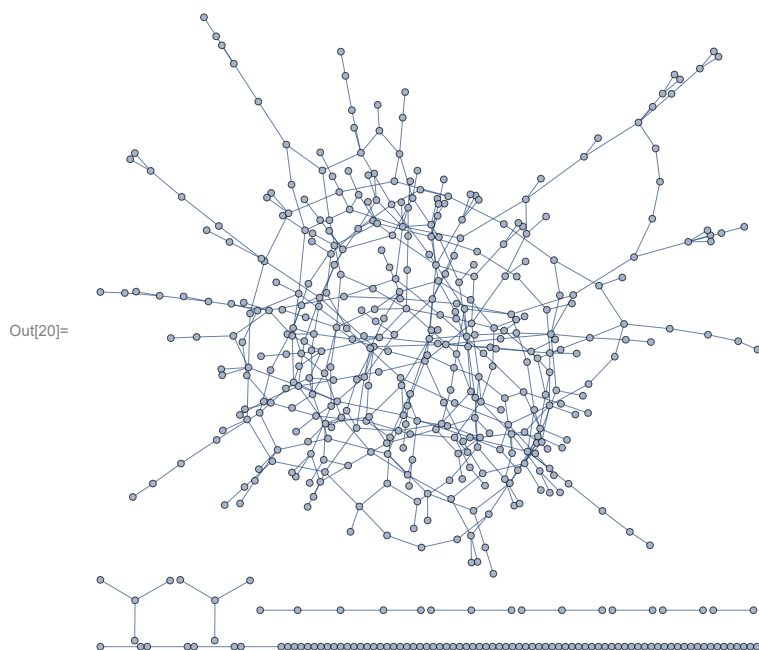
```
In[1]:= data1 = Import[NotebookDirectory[] <> "\\assoc.txt", "Data"];  
           импорт  директория файла блокнота  
  
In[2]:= e11 = Table[data1[[i, 1]] ↔ data1[[i, 2]], {i, 1, Length[data1]}];  
           таблица значений                                длина  
  
In[3]:= g1 = Graph[e11];  
           граф  
  
In[4]:= data2 = Import[NotebookDirectory[] <> "\\bio-celegans.txt", "Data"];  
           импорт  директория файла блокнота  
  
In[5]:= e12 = Table[data2[[i, 1]] ↔ data2[[i, 2]], {i, 1, Length[data2]}];  
           таблица значений                                длина  
  
In[6]:= g2 = Graph[e12];  
           граф  
  
In[7]:= data3 = Import[NotebookDirectory[] <> "\\bio-diseasome.txt", "Data"];  
           импорт  директория файла блокнота  
  
In[8]:= e13 = Table[data3[[i, 1]] ↔ data3[[i, 2]], {i, 1, Length[data3]}];  
           таблица значений                                длина  
  
In[9]:= g3 = Graph[e13];  
           граф  
  
In[10]:= data4 = Import[NotebookDirectory[] <> "\\ca-AstroPh.txt", "Data"];  
           импорт  директория файла блокнота  
  
In[11]:= e14 = Table[data4[[i, 1]] ↔ data4[[i, 2]], {i, 1, Length[data4]}];  
           таблица значений                                длина  
  
In[12]:= g4 = Graph[e14];  
           граф  
  
In[13]:= data5 = Import[NotebookDirectory[] <> "\\facebook.txt", "Data"];  
           импорт  директория файла блокнота  
  
In[14]:= e15 = Table[data5[[i, 1]] ↔ data5[[i, 2]], {i, 1, Length[data5]}];  
           таблица значений                                длина  
  
In[15]:= g5 = Graph[e15];  
           граф  
  
In[16]:= data6 = Import[NotebookDirectory[] <> "\\proteins.txt", "Data"];  
           импорт  директория файла блокнота  
  
In[17]:= e16 = Table[data6[[i, 1]] ↔ data6[[i, 2]], {i, 1, Length[data6]}];  
           таблица значений                                длина  
  
In[18]:= g6 = Graph[e16];  
           граф
```

1. Эволюция графа Эрдеша-Реньи и фазовый переход

Генерация случайного графа Эрдеша-Реньи заданного размера за заданное время

```
In[19]:= ErdősRenyiGraph[n_, t_] := Module[{maxe =  $\frac{n(n-1)}{2}$ , am = Table[0, {i, 1, n}, {j, 1, n}]},
      |программный модуль |таблица значений
      ne = Sort[RandomSample[Range[n], 2]], If[t > maxe, Print["Error: t>n(n-1)/2"],
      |сор... |случайная вы... |диапазон |условный оп... |печатать
      Do[While[am[[ne[[1]], ne[[2]]] != 0, ne = Sort[RandomSample[Range[n], 2]]];
      |... |цикл-пока |сор... |случайная вы... |диапазон
      am[[ne[[1]], ne[[2]]] = 1;
      am[[ne[[2]], ne[[1]]] = 1, {k, 1, t}];
      am]
```

```
In[20]:= AdjacencyGraph[ErdősRenyiGraph[500, 500]]
      |граф по матрице смежности
```



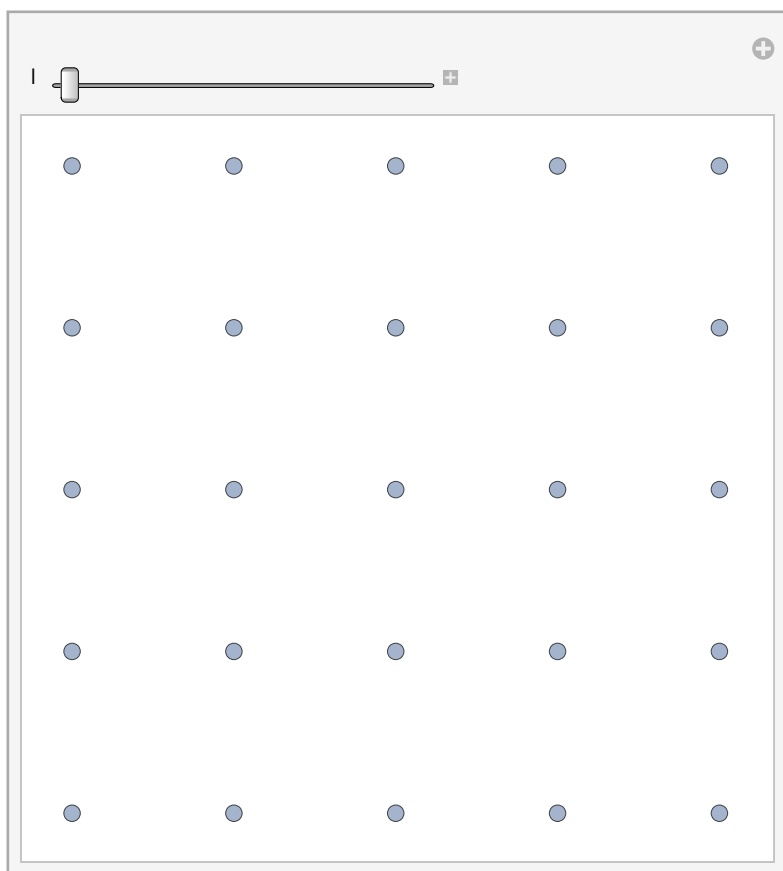
Генерация эволюции случайного графа Эрдеша-Реньи заданного размера за заданное время

```
In[21]:= ErdősRenyiGraphList[n_, t_] :=
Module[{maxe =  $\frac{n(n-1)}{2}$ , am = Table[0, {i, 1, n}, {j, 1, n}]},
  am1 = {}, ne = Sort[RandomSample[Range[n], 2]], AppendTo[am1, am];
  If[t > maxe, Print["Error: t>n(n-1)/2"],
  Do[While[am[[ne[[1]], ne[[2]]] != 0, ne = Sort[RandomSample[Range[n], 2]]];
  am[[ne[[1]], ne[[2]]] = 1;
  am[[ne[[2]], ne[[1]]] = 1;
  AppendTo[am1, am], {k, 1, t}];
  am1]
```

```
In[22]:= erg1 = ErdősRenyiGraphList[25, 25];
```

```
In[23]:= Manipulate[AdjacencyGraph[erg1[[1]], {1, 1, Length[erg1], 1}]
  | варьировать | граф по матрице смежности | длина
```

Out[23]=



Генерация случайного связного графа Эрдеша-Реньи заданного размера

```

In[24]:= ErdősRenyiConnectedGraph[n_] :=
Module[{am = Table[0, {i, 1, n}, {j, 1, n}], ne = Sort[RandomSample[Range[n], 2]]},
  |программный... |таблица значений |сор... |случайная вы... |диапазон
  While[ConnectedGraphQ[AdjacencyGraph[am]] ≠ True,
    |цикл... |связный граф? |граф по матрице смежности |истина
    While[am[[ne[[1]], ne[[2]]]] ≠ 0, ne = Sort[RandomSample[Range[n], 2]]];
    |цикл-пока |сор... |случайная вы... |диапазон
    am[[ne[[1]], ne[[2]]]] = 1;
    am[[ne[[2]], ne[[1]]]] = 1;
  am]

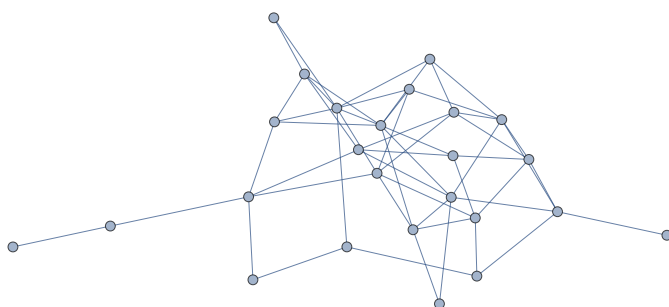
```

```

In[25]:= AdjacencyGraph[ErdősRenyiConnectedGraph[25]]
|граф по матрице смежности

```

Out[25]=



Генерация эволюции случайного графа Эрдеша-Реньи заданного размера до связного состояния

```

In[26]:= ErdősRenyiConnectedGraphList[n_] := Module[{am = Table[0, {i, 1, n}, {j, 1, n}],
  |программный... |таблица значений
  am1 = {}, ne = Sort[RandomSample[Range[n], 2]]], AppendTo[am1, am];
  |сор... |случайная вы... |диапазон |добавить в конец к
  While[ConnectedGraphQ[AdjacencyGraph[am]] ≠ True,
    |цикл... |связный граф? |граф по матрице смежности |истина
    While[am[[ne[[1]], ne[[2]]]] ≠ 0, ne = Sort[RandomSample[Range[n], 2]]];
    |цикл-пока |сор... |случайная вы... |диапазон
    am[[ne[[1]], ne[[2]]]] = 1;
    am[[ne[[2]], ne[[1]]]] = 1;
    AppendTo[am1, am];
    |добавить в конец к
  am1]

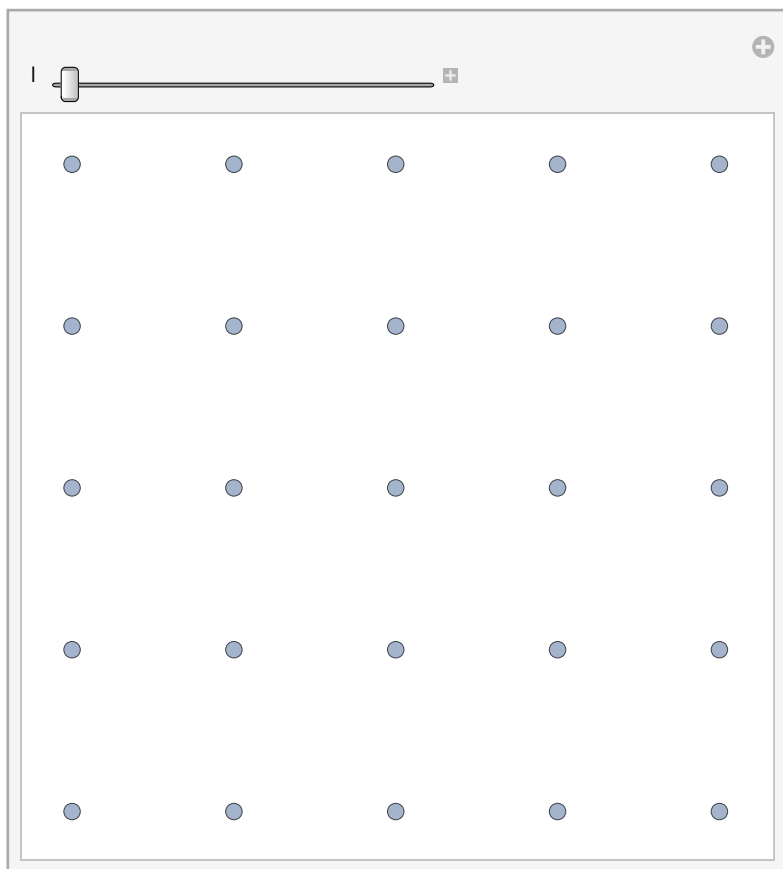
```

```

In[27]:= ercgl = ErdősRenyiConnectedGraphList[25];

```

In[28]:= **Manipulate**[**AdjacencyGraph**[**ercg1**[[1]], {1, 1, **Length**[**ercg1**], 1}],
 варьировать | граф по матрице смежности | длина



Out[28]=

Генерация эволюции размеров связных компонент случайного графа Эрдеша-Реньи заданного размера за заданное время

In[29]:= **ConnectedComponentsErdősRenyiGaph**[**n_**, **t_**] :=
Module[{**am** = **Table**[0, {**i**, 1, **n**}, {**j**, 1, **n**}}, **gc** = **Table**[1, {**i**, 1, **n**}},
 |программный... |таблица значений |таблица значений
gc1 = {}, **ne** = **Sort**[**RandomSample**[**Range**[**n**], 2]], **AppendTo**[**gc1**, **gc**];
 |сор... |случайная вы... |диапазон |добавить в конец к
Do[**While**[**am**[[**ne**[[1]], **ne**[[2]]] ≠ 0, **ne** = **Sort**[**RandomSample**[**Range**[**n**], 2]]];
 |... |цикл-пока |сор... |случайная вы... |диапазон
am[[**ne**[[1]], **ne**[[2]]] = 1;
am[[**ne**[[2]], **ne**[[1]]] = 1;
gc = **Map**[**Length**, **ConnectedComponents**[**AdjacencyGraph**[**am**]], {1}];
 |п... |длина |связные компоненты |граф по матрице смежности
AppendTo[**gc1**, **gc**], {**k**, 1, **t**}};
 |добавить в конец к
gc1]

График эволюции размеров связанных компонент случайного графа Эрдеша-Реньи

```
In[30]:= ConnectedComponentsErdősRenyiGaphPlot[sim_] :=
Module[{p0 = { $\frac{\text{Length}[sim]}{2}$ , sim[[1,  $\frac{\text{Length}[sim]}{2} + 1$ ]]},
программный модуль

p1 = {ReverseSortBy[{Range[Length[sim[[2]]], sim[[2]]}^T, Last][[1, 1]],
сортировка в обр... диап... длина последний
sim[[1, ReverseSortBy[{Range[Length[sim[[2]]], sim[[2]]}^T, Last][[1, 1] + 1]]},
сортировка в обр... диап... длина последний
p2 = {Length[sim], sim[[1, Length[sim] + 1]]},
длина длина

Show[ListLinePlot[Table[{Range[0, Length[sim[[1]] - 1, 1], sim[[i]]}^T, {i, 1, 5}],
пок... линейный гра... таблиц... диапазон длина

PlotRange → {{0, Length[sim[[1]]}, {0, Length[sim]}}, AspectRatio →  $\frac{\text{Length}[sim]}{\text{Length}[sim[[1]]]}$ ,
отображаемый диа... длина длина аспектное отношение

GridLines → {p0, p1, p2, {Length[sim[[1]] - 1, Length[sim]}}^T,
линии координатной сетки длина длина
Ticks → {p0, p1, p2, {Length[sim[[1]] - 1, Length[sim]}, {0, 0}}^T, PlotLabel →
деления длина длина пометка графика
"График изменения размера пяти самых больших связанных компонент в графе",
AxesLabel → {"Время", "Размер компоненты\n(Число вершин)"},
обозначения на осях
PlotLegends → PointLegend[{Red, Green, Blue},
легенды графика поточечная ле... кр... зелё... синий
{"Теоретическое время появления гигантской компоненты",
"Время начала убывания второй по размеру компоненты",
"Время равно общему числу вершин в графе"}, Joined → False],
соединё... ложь

Graphics[{PointSize[Medium], Red, Point[p0], Green, Point[p1], Blue, Point[p2]}],
графика размер то... средний кр... точка зелё... точка синий точка
ImageSize → Full]]
размер изоб... в полном объеме
```

Однократная генерация

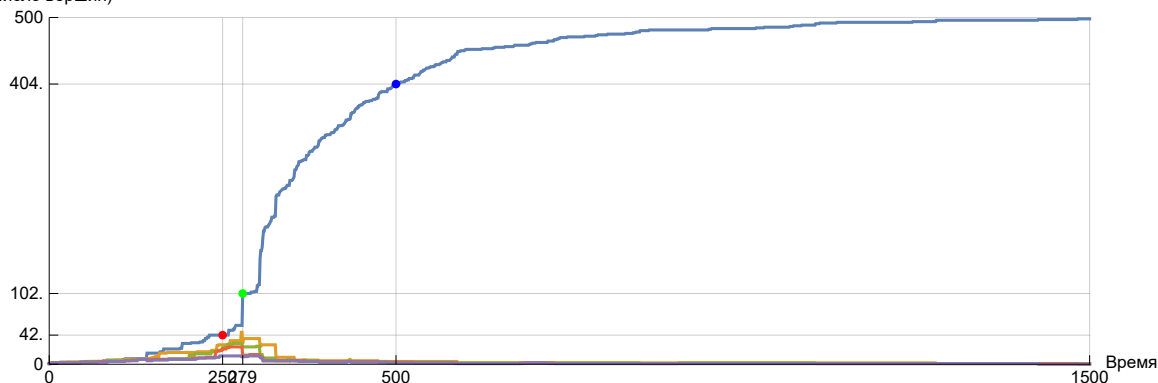
```
In[31]:= sim1 = Import[NotebookDirectory[] <> "\\sim1.m"];
импорт директория файла блокнота
(*sim1=N[PadRight[ConnectedComponentsErdősRenyiGaph[500,1500]]]^T;*)
... заполнить справа
```

In[32]:= ConnectedComponentsErdősRenyiGaphPlot[sim1]

График изменения размера пяти самых больших связных компонент в графе

Размер компоненты
(Число вершин)

Out[32]=



- Теоретическое время появления гигантской компоненты
- Время начала убывания второй по размеру компоненты
- Время равно общему числу вершин в графе

Среднее по 10 генерациям

In[33]:= sim10 = Import[NotebookDirectory[] <> "\\sim10.m"]; (*sim10=

импорт директория файла блокнота

Map[Mean, Table[N[PadRight[ConnectedComponentsErdősRenyiGaph[500, 1500]]], {1, 1, 10}]]^T,

п... сре... табл... .. заполнить справа

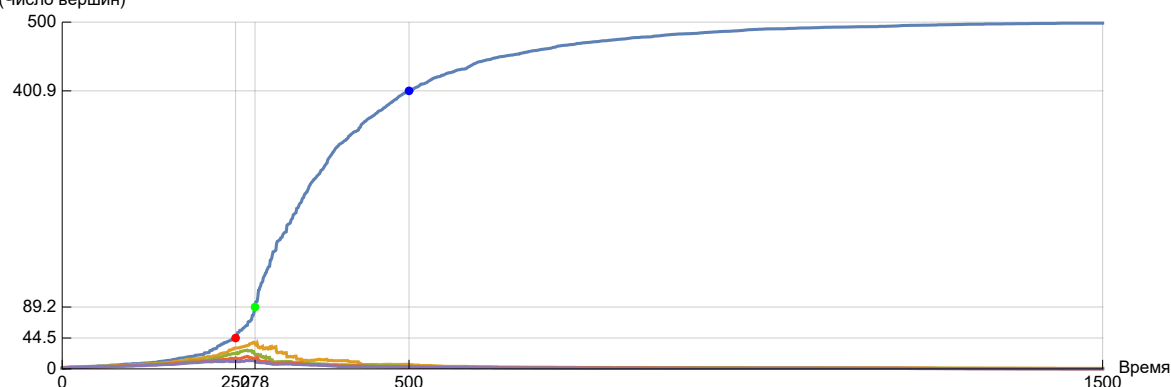
{1}]^T; *)

In[34]:= ConnectedComponentsErdősRenyiGaphPlot[sim10]

График изменения размера пяти самых больших связных компонент в графе

Размер компоненты
(Число вершин)

Out[34]=

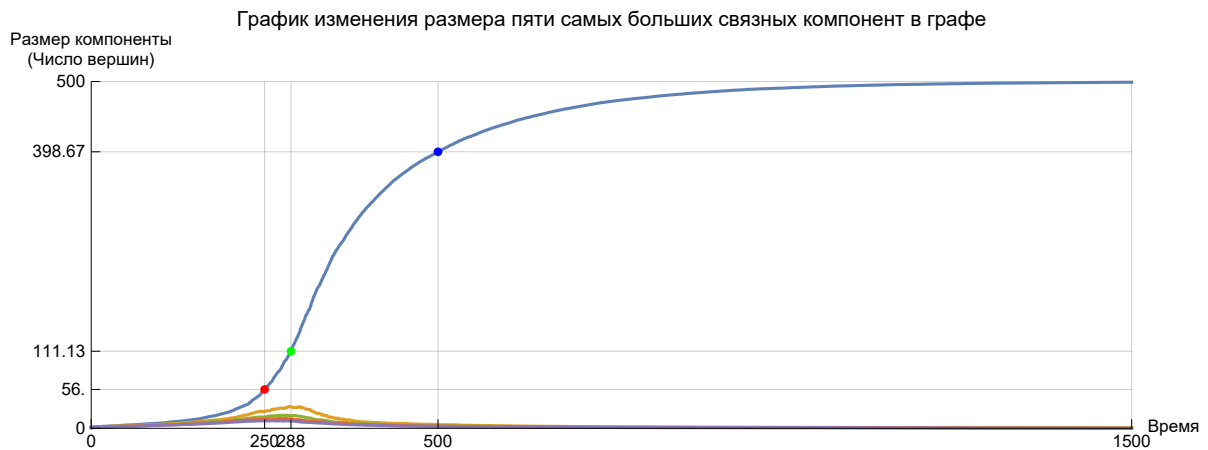


- Теоретическое время появления гигантской компоненты
- Время начала убывания второй по размеру компоненты
- Время равно общему числу вершин в графе

Среднее по 100 генерациям

```
In[35]:= sim100 = Import[NotebookDirectory[] <> "\\sim100.m"];
           импорт |директория файла блокнота
(*sim100=Map[Mean,Table[N[PadRight[ConnectedComponentsErdősRenyiGaph[500,1500]]],
           |п... |сре... |табл... |... |заполнить справа
           {1,1,100}]^T,{1}]^T;*)
```

```
In[36]:= ConnectedComponentsErdősRenyiGaphPlot[sim100]
```

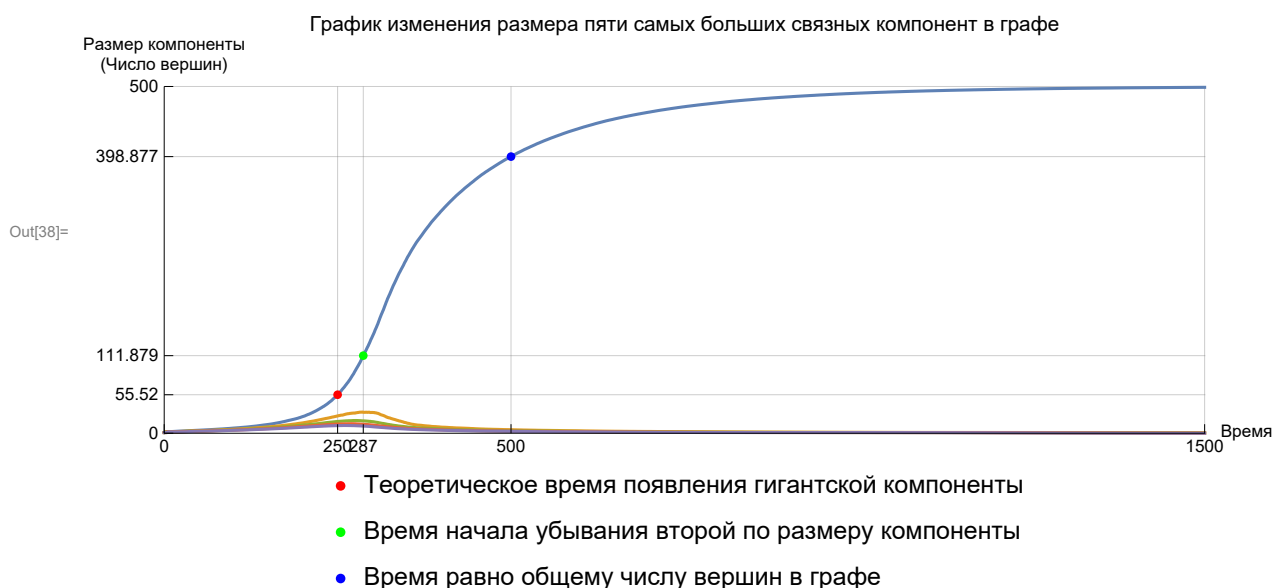


- Теоретическое время появления гигантской компоненты
- Время начала убывания второй по размеру компоненты
- Время равно общему числу вершин в графе

Среднее по 1000 генерациям

```
In[37]:= sim1000 = Import[NotebookDirectory[] <> "\\sim1000.m"];
           импорт |директория файла блокнота
(*sim1000=Map[Mean,Table[N[PadRight[ConnectedComponentsErdősRenyiGaph[500,1500]]],
           |п... |сре... |табл... |... |заполнить справа
           {1,1,1000}]^T,{1}]^T;*)
```


In[38]:= ConnectedComponentsErdősRenyiGaphPlot[sim1000]



2. Малый мир сложных сетей

Средний кратчайший путь в графе

```
In[39]:= MeanShortestPath[graph_] :=
Module[{out = {}, cgc = {}, cgcd = {}, cgcdt = {}}, If[ConnectedGraphQ[graph] == True,
  [программный модуль] [...] [связный граф?] [истина]
  out = N[
$$\frac{\text{Total}[\text{Flatten}[\text{GraphDistanceMatrix}[\text{graph}]]]}{\text{VertexCount}[\text{graph}]^2 - \text{VertexCount}[\text{graph}]}$$
],
  [численное приближение]
  cgc = ConnectedGraphComponents[graph];
  [связные граф-компоненты]
  cgcd = Table[GraphDistanceMatrix[cgc[[i]], {i, 1, Length[cgc]}], {i, 1, Length[cgc]}];
  [табл...] [матрица расстояний на графе] [длина]
  cgcdt =
  Table[If[cgcd[[i]] == {{0}}, {0, 0}, {N[
$$\frac{\text{Total}[\text{Flatten}[\text{cgcd}[[i]]]}{\text{VertexCount}[\text{cgc}[[i]]^2 - \text{VertexCount}[\text{cgc}[[i]]}$$
],
    [условный оператор] [численное приближение]
    Length[cgcd[[i]] (Length[cgcd[[i]] - 1)
    [длина]
    }], {i, 1, Length[cgcd]}];
  out = 
$$\frac{\text{Total}[\text{Table}[\text{cgcdt}[[i, 1]] \text{cgcdt}[[i, 2]], \{i, 1, \text{Length}[\text{cgcdt}]\}]]}{\text{Total}[\text{cgcdt}[[\text{All}, 2]]]}$$
;
  out]
```

Оценка среднего кратчайшего пути в графе

```
In[40]:= EstimatedAveragePath[graph_, numberofvertexpairs_] :=
Module[{maxnvp =  $\frac{\text{VertexCount}[\text{graph}] (\text{VertexCount}[\text{graph}] - 1)}{2}$ , gvl = VertexList[graph],
  программный модуль 2 список вершин графа

  eal = Table[0, {i, 1, numberofvertexpairs}], rvp = {}, rvp1 = {},
  таблица значений

  If[numberofvertexpairs > maxnvp, Print["Error: number of vertex pairs > n(n-1)/2"],
  печатать

  rvp = Sort[RandomSample[gvl, 2]];
  сор... случайная выборка

  Do[While[MemberQ[rvp1, rvp] == True || FindShortestPath[graph, rvp[[1]], rvp[[2]] == {},
  цикл... элемент списка? истина найти кратчайший путь

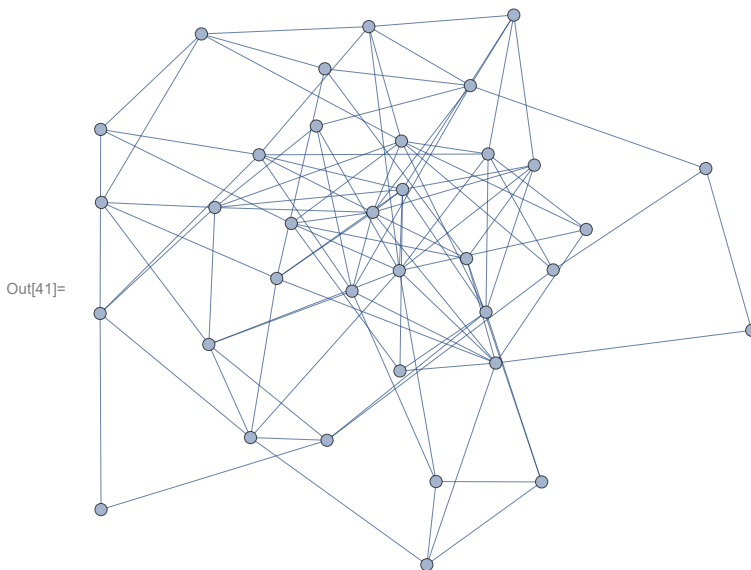
  rvp = Sort[RandomSample[gvl, 2]];
  сор... случайная выборка

  AppendTo[rvp1, rvp];
  добавить в конец к

  eal[[i]] = Length[FindShortestPath[graph, rvp[[1]], rvp[[2]]] - 1,
  длина найти кратчайший путь

  {i, 1, numberofvertexpairs}];
  N[Mean[eal]]]
  ... среднее значение
```

```
In[41]:= rg = RandomGraph[{35, 100}]
  случайный граф
```



```
In[42]:= EstimatedAveragePath[rg, 100]
```

Out[42]= 2.27

```
In[43]:= EstimatedAveragePath[rg, 595]
```

Out[43]= 2.19664

```
In[44]:= MeanShortestPath[rg]
```

```
Out[44]= 2.19664
```

Мера малого мира

```
In[45]:= Sigma[graph_] :=
```

```
Module[{c = Mean[LocalClusteringCoefficient[graph]], cr =  $\frac{\text{Mean}[\text{VertexDegree}[\text{graph}]]}{\text{VertexCount}[\text{graph}]}$ ,  
[программны... [сре... [коэффициент локальной кластеризации
```

```
l = MeanShortestPath[graph], lr =  $\frac{\text{Log}[\text{VertexCount}[\text{graph}]] - \text{EulerGamma}}{\text{Log}[\text{Mean}[\text{VertexDegree}[\text{graph}]]]} + \frac{1}{2}$ },  $\frac{c \text{ lr}}{cr \text{ l}}$ ]
```

Параметры графа

```
In[46]:= GraphParameters[graph_, pairscount_] :=
```

```
{VertexCount[graph], EdgeCount[graph], EstimatedAveragePath[graph, pairscount],  
[число вершин [число рёбер
```

```
MeanShortestPath[graph], N[Log[VertexCount[graph]]], Sigma[graph]}  
[... [на... [число вершин
```

```

In[47]:= Grid[Prepend[{Prepend[GraphParameters[g1, 100], "assoc"],
  таб... [добавить ... [добавить в начало
    Prepend[GraphParameters[g2, 100], "bio-celegans"],
    [добавить в начало
    Prepend[GraphParameters[g3, 100], "bio-diseasome"],
    [добавить в начало
    Prepend[GraphParameters[g4, 100], "ca-AstroPh"],
    [добавить в начало
    Prepend[GraphParameters[g5, 100], "facebook"],
    [добавить в начало
    Prepend[GraphParameters[g6, 100], "proteins"]}],
  {"Name", "Vertex count", "Edge count", "Estimated average path",
    "Mean shortest path", "ln(vertex count)", "Small-world measure"}],
  [среднее значение [малый
  Background → {None, {{{Pink, Lighter[Blue, 0.7]}}, {1 → Gray}}},
  [фон [ни одног... [розо... [более с... [синий [серый
  Dividers → All, Spacings → {1, 1}
  [разделители [всё [размер зазора

```

Name	Vertex count	Edge count	Estimated average path	Mean shortest path	ln(vertex count)	Small-world measure
assoc	6437	36 921	3.76	3.80954	8.76982	70.2352
bio-celegans	453	2025	2.71	2.66379	6.11589	37.2392
bio-diseasome	516	1188	6.46	6.50899	6.24611	46.1102
ca-AstroPh	18 771	198 050	4.17	4.19399	9.84007	473.183
facebook	4039	88 234	3.67	3.69251	8.30375	38.5921
proteins	335	1792	4.79	4.82131	5.81413	11.4945

3. Устойчивость малого мира сложных сетей

Удаление заданного процента ребер в графе

```

In[48]:= DeletePercentEdges[graph_, percent_] :=
  EdgeDelete[graph, RandomSample[EdgeList[graph], Round[EdgeCount[graph] (
    [удалить ребро [случайная вы... [список рёбер [окру... [число рёбер
     $\left(\frac{\text{percent}}{100}\right)]]]]$ 
```

Сравнение параметров графа и графа с уменьшенным числом ребер

```

In[49]:= ChangeGraphParameters[graph_, percent_] :=
  Module[{mg = DeletePercentEdges[graph, percent]}, {MeanShortestPath[graph],
  [программный модуль
    MeanShortestPath[mg], N[Mean[LocalClusteringCoefficient[graph]]],
    [... [сре... [коэффициент локальной кластеризации
    N[Mean[LocalClusteringCoefficient[mg]]], Sigma[graph], Sigma[mg]]]
    [... [сре... [коэффициент локальной кластеризации

```

```

In[50]:= Grid[Prepend[{Prepend[ChangeGraphParameters[g1, 10], "assoc"],
  Prepend[ChangeGraphParameters[g2, 10], "bio-celegans"],
  Prepend[ChangeGraphParameters[g3, 10], "bio-diseasome"],
  Prepend[ChangeGraphParameters[g4, 10], "ca-AstroPh"],
  Prepend[ChangeGraphParameters[g5, 10], "facebook"],
  Prepend[ChangeGraphParameters[g6, 10], "proteins"]},
  {"Name", "Mean shortest path", "Mean shortest path (-10%)",
    "Mean Clustering Coefficient", "Mean Clustering Coefficient (-10%)",
    "Small-world measure", "Small-world measure (-10%)"},
  Background → {None, {{{Pink, Lighter[Blue, 0.7]}}, {1 → Gray}}},
  Dividers → All, Spacings → {1, 1}]

```

Name	Mean shortest path	Mean shortest path (-10%)	Mean Clustering Coefficient	Mean Clustering Coefficient (-10%)	Small-world measure	Small-world measure (-10%)
assoc	3.80954	3.884	0.123601	0.107829	70.2352	69.3985
bio-celegans	2.66379	2.73836	0.646463	0.56985	37.2392	36.9625
bio-diseasome	6.50899	6.79761	0.63583	0.530088	46.1102	43.584
ca-AstroPh	4.19399	4.29093	0.630627	0.553445	473.183	464.847
facebook	3.69251	3.88242	0.605547	0.538331	38.5921	37.0915
proteins	4.82131	5.00884	0.653178	0.580317	11.4945	11.3346

Out[50]=