


**Нижегородский государственный университет им. Н.И. Лобачевского**

**Институт информационных технологий, математики и механики**

**Нижегородский государственный университет им. Н.И. Лобачевского**

**Институт информационных технологий, математики и механики**

***ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА ГРАФОВ***

# Лекция 2. Моделирование графов

Пирова А.Ю.
Кафедра МОСТ

# Содержание

- Прикладные области
- Свойства сетей
- Коллекции графов прикладных областей
- Зачем моделировать графы?
- Классификация генераторов
- Некоторые модели графов:
  - Модель Эрдеша – Реньи
  - Модель Барабаши – Альберт
  - Модель R-MAT
  - Модель графов Кронекера
- Генераторы графов
- Заключение

# Прикладные области

- Где возникают графы большого порядка?
  - Транспортные сети
  - Коммуникационные сети
  - Веб-графы, сети цитирований
  - Социальные сети, сети сообществ
  - Энергетические системы
  - Биологические системы (взаимодействие между белками, метаболические, нейронные сети, пищевые цепочки)
  - Бизнес-процессы, кибербезопасность
  - Проектирование микросхем…

# Коллекции графов

- Suite Sparse Matrix Collection (бывшая The University of Florida Sparse Matrix Collection) https://sparse.tamu.edu/
Матрицы из различных прикладных областей, включая 2D и 3D дискретизации различных процессов, задачи оптимизации, сети дорог.
- Stanford Large Network Dataset Collection https://snap.stanford.edu/data/ (включено в Suite Sparse)
Графы социальных сетей, веб-графы, сети дорог
- The Koblenz Network Collection http://konect.cc/
Сети из разных областей: социальные, веб, физические, сети коммуникаций и др. (*даны свойства графов)
- Обзор коллекций сетей Марка Ньюмена http://www-personal.umich.edu/~mejn/netdata/

# Коллекции графов

- Graph Challenge MIT Data Sets
  https://graphchallenge.mit.edu/data-sets
  Коммуникационные и транспортные сети, веб графы, графы Кронекера
- Коллекции данных конкурсов DIMACS Implementation Challenges
  – Сети дорог http://users.diag.uniroma1.it/challenge9/download.shtml
  – Коллекция графов для разделения, включена в Suite Sparse https://www.cc.gatech.edu/dimacs10/downloads.shtml
- Network Repository http://networkrepository.com/networks.php
  Сети из различных областей: биологические, транспортные, коммуникационные и др.

# МОДЕЛИРОВАНИЕ ГРАФОВ

# Необходимость моделирования графов

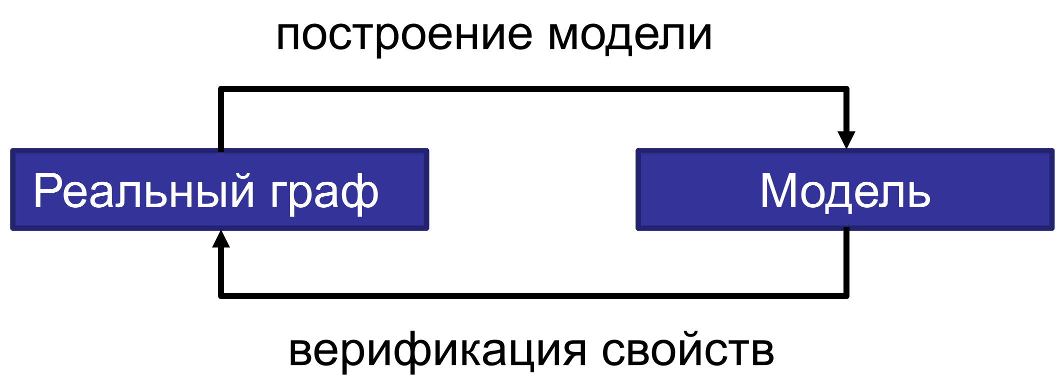
- Зачем моделировать графы?
  – **Создание графов для тестирования алгоритмов**. Цель – получить набор графов с заданными свойствами, но отличающимися друг от друга в некотором смысле; либо получить набор графов с заданными свойствами различных размеров. Реальные графы могут существовать в единичном экземпляре.
  – **Выборка графа (graph sampling)**. Цель – создать граф для моделирования, воспроизводящий свойства реального, если реальный граф слишком большой (например, граф социальной сети).
  – **Экстраполяция графа**. Цель – построение графа большего размера, чем реальный, для предсказания эволюции графа в будущем или проверки его свойств на больших размерах.

# Необходимость моделирования графов

- Зачем моделировать графы?
  – **Сжатие графа**. Цель – создать граф схожей структуры, удовлетворяющий ограничениям по памяти вычислительной системы, либо с целью визуализации.
  – **Анонимизация графа**. Цель – создать граф, сохраняющий важные свойства оригинала, но достаточно от него отличающийся, чтобы обеспечить конфиденциальность данных. Реальный граф может содержать приватную информацию о здоровье, персональных данных, контактах человека.

# Необходимость моделирования графов

- Модельный граф должен воспроизводить свойства и структуру исходного графа
- Цикл моделирования:



# Классификация генераторов

- Нет единой общепринятой классификации моделей и генераторов сетей
- Классификация генераторов Чакрабати и Фалустос:

Генераторы графов

- Генераторы случайных графов
- Генераторы на основе предпочтительного присоединения
- Генераторы на основе оптимизации
- Географические модели
- Интернет-специфичные модели

# Модель Эрдеша–Реньи

- Первая модель случайного графа, предложена в 1959 г.
- Обозначается $G(n, M)$ или $G(n, p)$
- **Модель $\boldsymbol{G(n,p)}$.** Пусть множество вершин графа $V_n = \{1, 2, \dots, n\}$. Полный граф $K_n$ содержит $M = C_n^2$ ребер. В случайном графе $G(V_n, E)$ любые две вершины $i$ и $j$ соединяются ребром $e \in E$ с вероятностью $p \in [0,1]$ не зависимо от остальных $C_n^2 - 1$ пар вершин (схема Бернулли).
- **Модель $\boldsymbol{G(n,M)}$**. Граф выбирается равновероятно и случайно из всех графов, имеющих $n$ вершин и $M$ ребер.

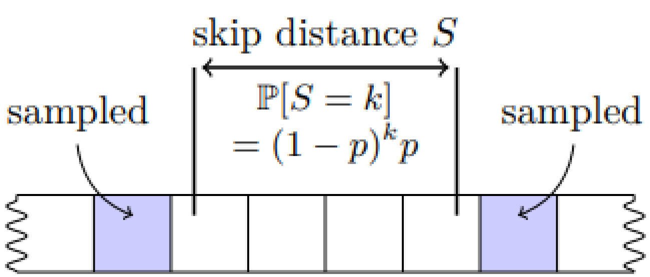
- Графы хорошо теоретически изучены.

# Модель Эрдеша–Реньи

- Некоторые свойства:
  - Если $p = \frac{1}{2}$, то все графы с $n$ вершинами равновероятны.
  - Пусть $p = \frac{c \ln n}{n}$. Если $c > 1$, то почти всегда случайный граф связен. Если $c < 1$, то почти всегда случайный граф не связный.
  - Пусть $p = \frac{c}{n}$. Тогда при любом $c < 1$ существует такая константа $\beta$ $\beta = \beta(c)$, что почти наверное каждая компонента случайного графа имеет не более $\beta \ln n$ вершин. При любом $c > 1$ существует такая константа $\gamma = \gamma(c) \in (0, 1)$, что почти наверное среди компонент случайного графа есть одна (гигантская), число вершин которой не меньше $\gamma n$.

# Модель Эрдеша–Реньи. Создание выборки

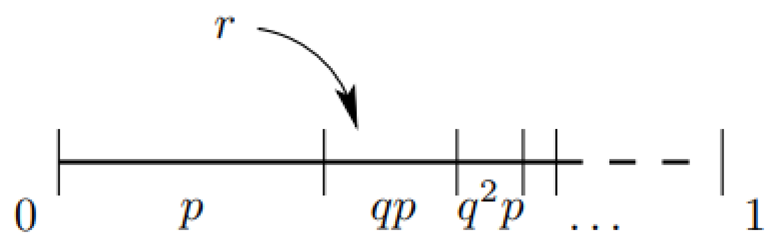
- Для генерации необходимо создание выборки из множества всех ребер по схеме Бернулли (sampling).
- Геометрический подход:
  - Для каждого ребра вероятность того, что он будет следующим на $k$-й итерации равна $\frac{(1-p)^{k-1}}{p}$. Процесс выбора ребер можно рассмотреть как выбор интервалов-«пропусков» между ними.
  - Интервалы между выбираемыми элементами имеют геометрическое распределение с параметром $1/p$.




Penschuck M. et al. Recent Advances in Scalable Network Generation //arXiv preprint arXiv:2003.00736. – 2020.

# Модель Эрдеша–Реньи. Создание выборки

- Геометрический подход:
  - Каждой итерации $k$ ставим в соответствие интервал $I_k \subseteq [0, 1)$ длины $(1 - p)^{k-1}p$. Тогда промежуток между выбираемыми элементами вычисляется, как наименьшее $k$, для которого интервал $I_k$ заканчивается после случайно выбранного $r \in [0, 1)$.




$$r < 1 - (1 - p)^k \leftrightarrow k > \frac{\log(1 - r)}{\log(1 - p)}$$

# Модель Эрдеша–Реньи. Алгоритм

- Алгоритм (Батаджели, Брандес). Вычислительная сложность $O(n+m)$ для графа с $n$ вершинами и $m$ ребрами

**ALG. 1**: $\mathcal{G}(n,p)$

**Input**: number of vertices $n$, edge probability $0 < p < 1$

**Output**: $G = (\{0,\dots,n-1\}, E) \in \mathcal{G}(n,p)$

$E \leftarrow \emptyset$

$v \leftarrow 1;\quad w \leftarrow -1$

**while** $v < n$ **do**

  draw $r \in [0,1)$ uniformly at random

  $w \leftarrow w + 1 + \left\lfloor \dfrac{\log(1-r)}{\log(1-p)} \right\rfloor$

  **while** $w \geq v$ **and** $v < n$ **do**

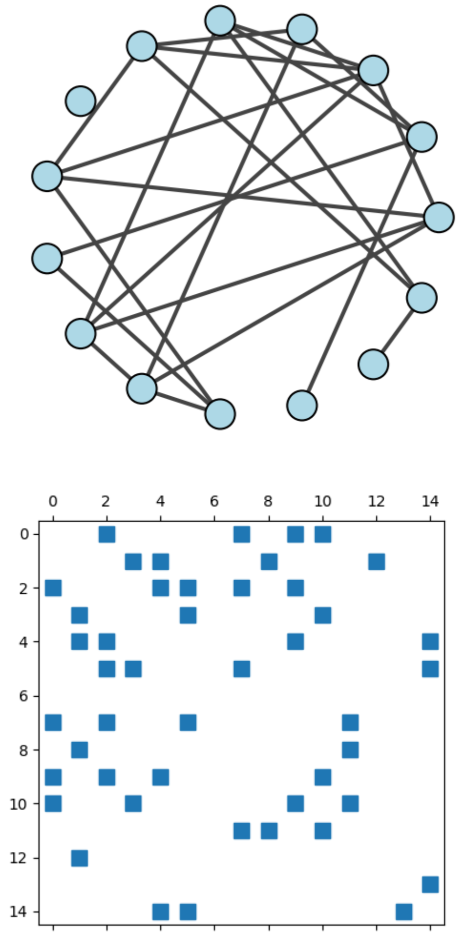
    $w \leftarrow w - v;\ v \leftarrow v + 1$

  **if** $v < n$ **then** $E \leftarrow E \cup \{v, w\}$

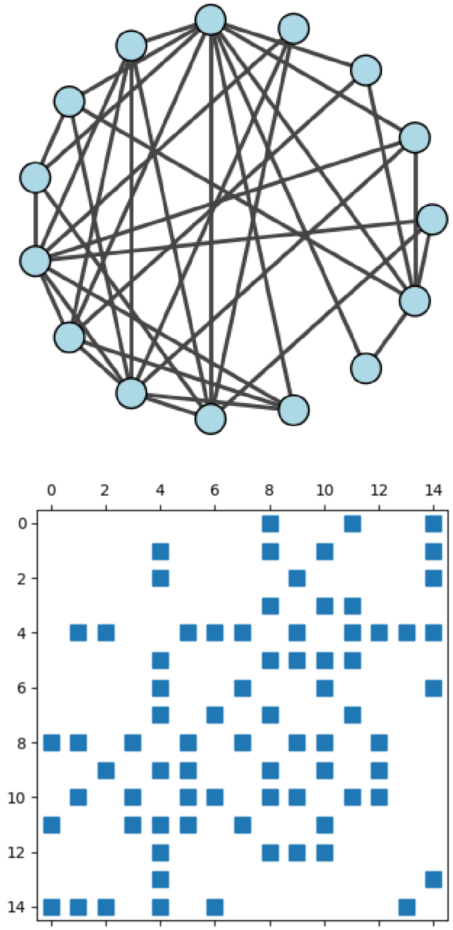
Выборка ребра из верхнего треугольника матрицы.

# Модель Эрдеша–Реньи. Примеры

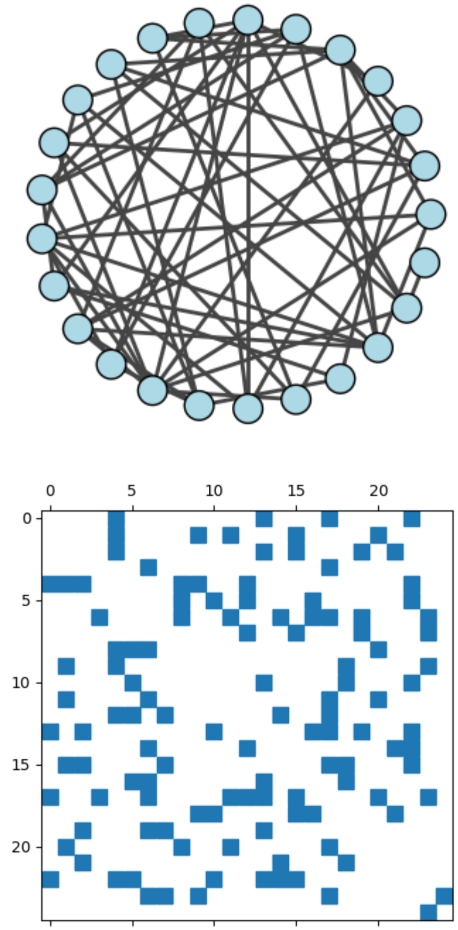
n = 15, p = 0.2



n = 15, p = 0.3



n = 25, p = 0.2

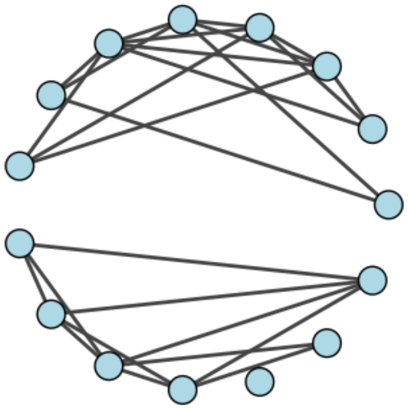


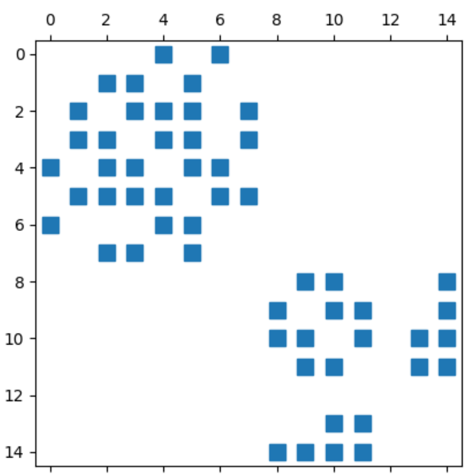
# Случайный геометрический граф

- Случайный геометрический граф (Random geometric graph, RGG) – ненаправленный граф, полученный случайным размещением $N$ вершин на некотором метрическом пространстве (чаще всего, на $d$-мерном кубе $[0,1)^d$). Ребро $e = (v_1, v_2)$ добавляется в граф, если расстояние между $v_1$ и $v_2$ менее $r \in (0,1)$. Чаще всего используется Евклидово расстояние.

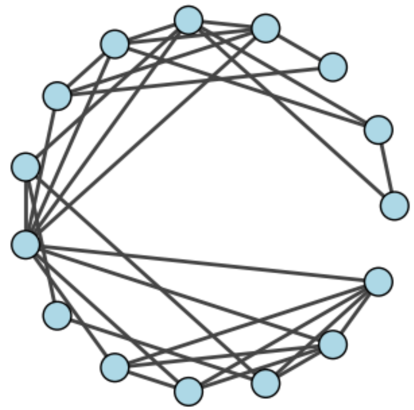
# Случайный геометрический граф. Примеры

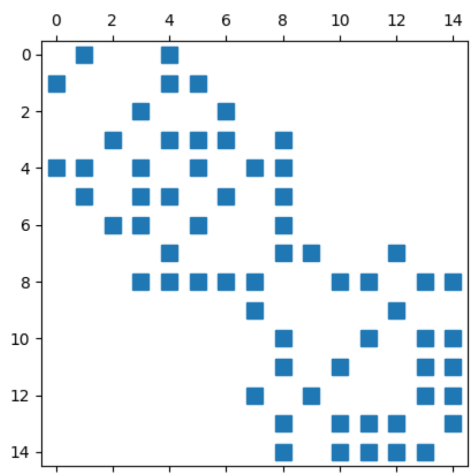
n = 15, r = 0.3



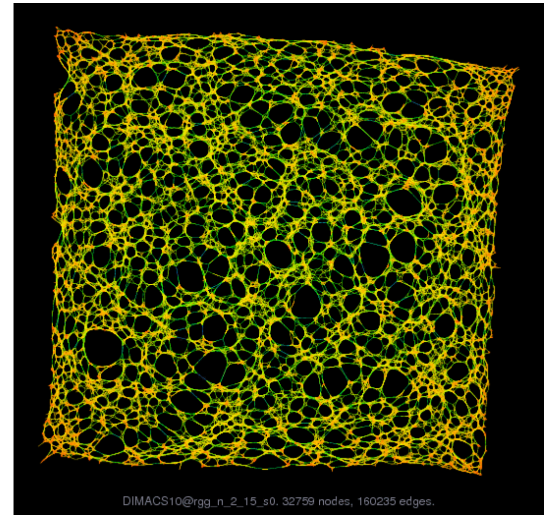


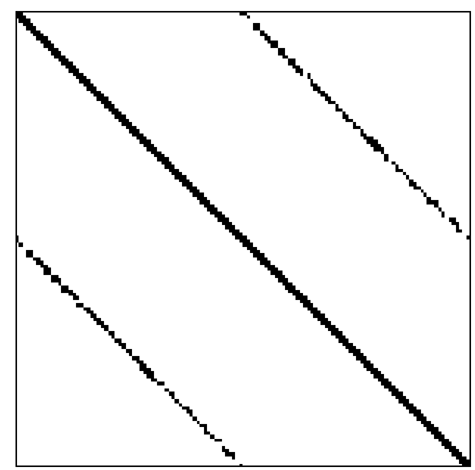

n = 15, r = 0.4






n = 32 768, m = 320 480



# Свойства сетей

- Особые свойства у графов сетей (Internet, веб, граф цитат, распределение сообществ и др).
- Степени вершин распределены согласно степенной функции
  - Для графа с числом вершин $N_d$ и степенью вершин $d$ справедливо $N_d \propto d^{-\gamma}, \gamma \geq 1$.
  - Такие графы называют ***безмасштабными*** **(scale-free)**
- Малый диаметр (феномен «малого мира»).
  - *Эффективный диаметр* – минимальное число ребер (связей), которыми соединено большинство пар вершин.
  - В 1999 г. эффективный диаметр Web был равен 5–7.
- Степенной закон уплотнения (densification power law). Число ребер $m$ растет пропорционально степени от числа вершин: $m(t) \sim \ n(t)^{\alpha}, 1 < \alpha < 2$.

# Свойства сетей

- Достижимость вершин (hop-plot):
  - Для заданной длины пути h вычисляется, какая доля всех пар вершин графа находится на расстоянии не более, чем h друг от друга.
  - Граф Интернета имеет степенной характер зависимости достижимости вершин: $N(h) \sim h^{\alpha}$ для $h \ll d$
- Высокий коэффициент кластеризации
- Сокращение диаметра (shrinking diameter). При росте числа узлов эффективный диаметр сети сокращается или стабилизируется
- Сети со временем становятся плотнее, то есть число ребер растет быстрее, чем число вершин

# Предпочтительное присоединение. Модель Барабаши–Альберт

- *Принцип предпочтительного присоединения*: Пусть дан стартовый граф из $n_0$ вершин. В процессе генерации новые вершины $v_i, i = n_0 + 1, \dots, n$ присоединяются к $d$ существующим, выбранным по определенному признаку. Как правило, признак – степень вершины.
- Принцип используется в модели Барабаши–Альберт и ее модификациях, модели копирования и др.
- Барабаши–Альберт: $P(\text{ребра к вершине } v) = \frac{d(v)}{\sum_i d(v_i)}$

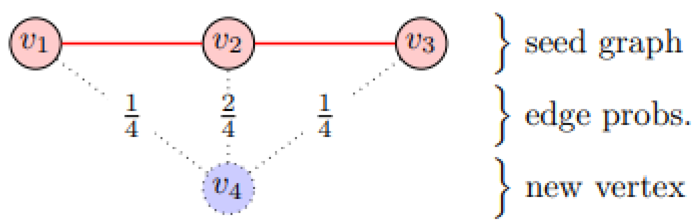
  $d(v_i)$ – степень вершины
- Дороговцев и др.: $P(\text{ребра к вершине } v) = \frac{\text{A}+d(v)}{\sum_i (A+d(v_i))}, A \geq 0$

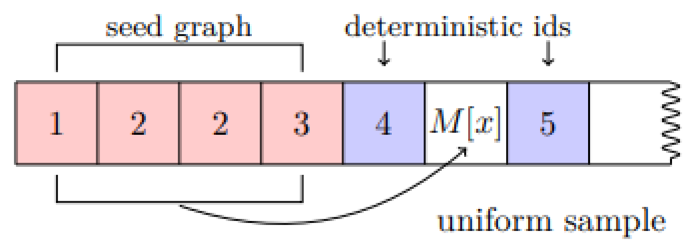
# Предпочтительное присоединение. Модель Барабаши–Альберт

- Свойства модели:
  - Распределение степеней вершин соответствует степенному закону не зависимо от числа ребер $P \approx k^{-3}$.
  - Для функции Дороговцева $P \approx k^{-\gamma}, \gamma = 2 + \frac{A}{\Delta m}, \in [2, \infty)$, где $\Delta m$ - это количество новых ребер, добавляемых на каждом шаге.
  - Отражает свойство «малого мира»: при больших $N$ диаметр графа растет как $O(\log N)$ для $d = 1$ и $O(\frac{\log N}{\log \log N})$ для $d \geq 2$.
  - Модель не воспроизводит сообщества
  - Полученный граф ненаправленный и состоит из одной компоненты

# Модель Барабаши–Альберт. Алгоритм

- Алгоритм (Батаджели, Брандес). Вычислительная сложность $O(n+m)$ для графа с $n$ вершинами и $m$ ребрами
- Используется вспомогательный массив $M$ размера $2*d*n$, где $d$ – число одновременно создаваемых ребер (параметр). $M[2i]$ и $M[2i+1]$ хранят концы ребра $e_i$. Число появлений вершины $v$ в массиве $M$ является ее степенью. $M[k] \leq \lfloor k/2 \rfloor$
- Создание нового ребра: для новой вершины $v$ смежную вершину будем выбирать по случайному индексу из массива $M$. Генерируем $r \in \{0, \ldots, 2dv+i\}$ для $i$-го соседа. Тогда новое ребро $(v, M[r])$




Penschuck M. et al. Recent Advances in Scalable Network Generation //arXiv preprint arXiv:2003.00736. – 2020.

# Модель Барабаши–Альберт. Алгоритм

**ALG. 5**: preferential attachment

**Input**: number of vertices $n$
minimum degree $d \geq 1$

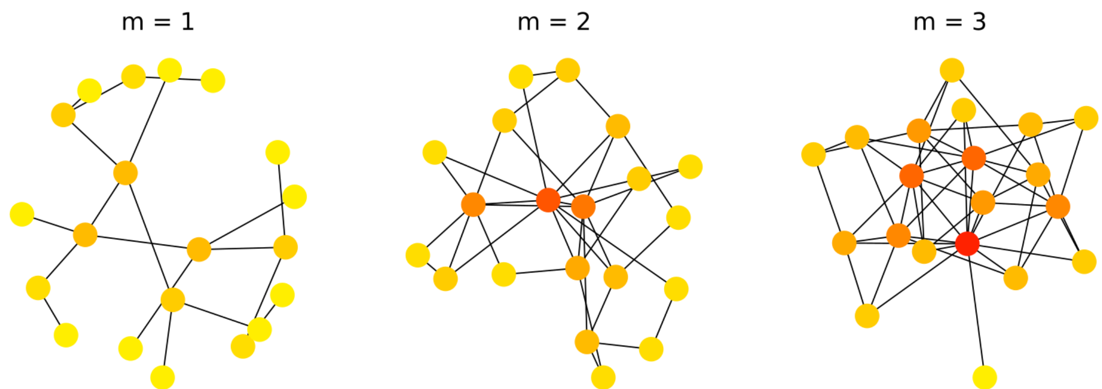
**Output**: scale-free multigraph
$G = (\{0, \ldots, n-1\}, E)$

$M$: array of length $2nd$
**for** $v = 0, \ldots, n-1$ **do**
  **for** $i = 0, \ldots, d-1$ **do**
    $M[2(vd+i)] \leftarrow v$
    draw $r \in \{0, \ldots, 2(vd+i)\}$ uniformly at random
    $M[2(vd+i)+1] \leftarrow M[r]$
$E \leftarrow \emptyset$
**for** $i = 0, \ldots, nd-1$ **do**
  $E \leftarrow E \cup \{M[2i], M[2i+1]\}$

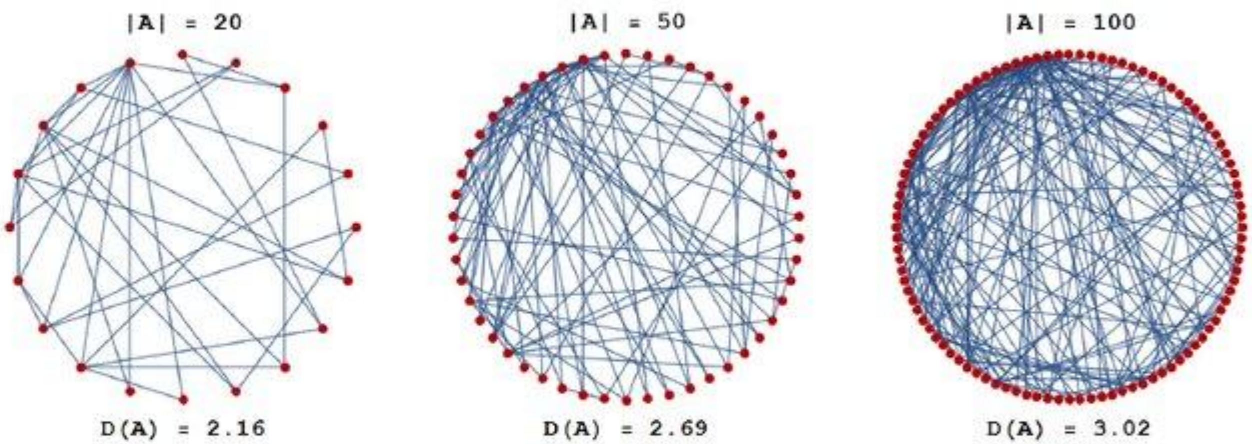
Выбор ребра. Возможны петли и повторы ребер.

- Как избежать повторяющихся ребер?

# Модель Барабаши–Альберт. Пример

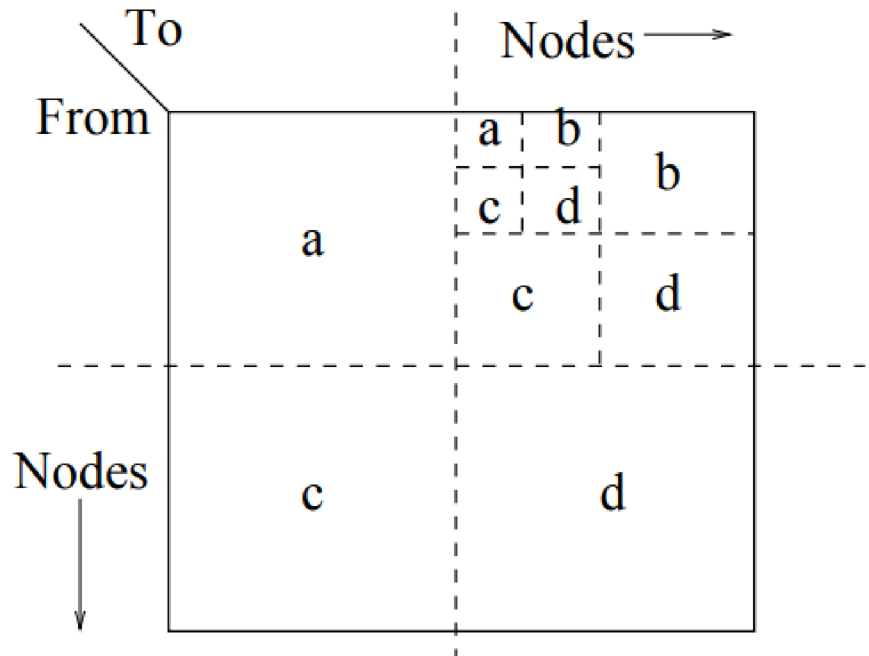



https://en.wikipedia.org/wiki/Barab%C3%A1si%E2%80%93Albert_model



# Модель R-MAT

- Модель Recursive Matrix (R-MAT) позволяет сгенерировать граф со степенным распределением степеней вершин.
- Идея: рекурсивно разделить матрицу смежности на 4 равные части, распределяя ребра между этими частями с неравной вероятностью $a, b, c, d$ $(a + b + c + d = 1)$.
- При определении места очередного ребра каждая часть матрицы выбирается с заданной вероятностью. Выбранная область также разделяется на 4 части. Процедура выполняется рекурсивно до тех пор, пока не будет получена часть размером 1 * 1. На этом месте будет ребро.



# Модель R-MAT

- Число вершин - $2^n$, $n$ – *степень* графа. Если хотим смоделировать граф с $N$ вершинами, то $n = \lceil \log_2 N \rceil$.
- При генерации ребра могут дублироваться. В итоговой матрице смежности повторные значения не учитываются.
- Полученный граф будет направленным (матрица смежности несимметрична). Как правило, $a \geq b, a \geq c, a \geq d$
- Ненаправленный граф: установить $b = c$, после генерации ребер отбросить значения над верхней диагональю матрицы и скопировать в нее значения из нижнего треугольника матрицы.

# Графы Кронекера

- Цель: построить графы, которые отражают свойство реальных сетей – с увеличением размера становятся плотнее при постоянном или сокращающемся диаметре.
- Идея: рекурсивно строить последовательность самоподобных графов, используя произведение Кронекера матриц.
- Пусть граф-инициатор $G_1$ содержит $N_1$ вершину и $E_1$ ребро. Тогда построим последовательность графов $G_2, G_3, \dots,$ таких, что $k$-й граф содержит $N_k = N_1^k$ вершин и $E_k = E_1^k$ ребер.

# Графы Кронекера

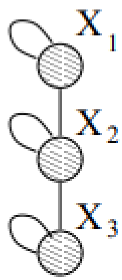
- Произведение Кронекера двух матриц $A$ размера $n \times m$ и $B$ размера $n' \times m'$ – матрица $C$ размером $(n * n') \times (m * m')$ :

$$C = A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1m}B \\ a_{21}B & a_{22}B & \cdots & a_{1m}B \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1}B & a_{n2}B & \ldots & a_{nm}B \end{pmatrix}$$

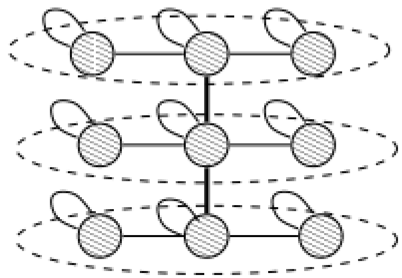
- Произведение Кронекера двух графов $G$ и $H$ (Вейчел, 1962) = произведение Кронекера их матриц смежности $A(G)$ и $A(H)$
- Ребро $(X_{ij}, X_{kl}) \in G \otimes H$ тогда и только тогда, когда $(X_i, X_k) \in G$ и $(X_j, X_l) \in H$.
- Последовательность графов порождается многократным выполнением произведения Кронекера над исходным графом:

$$G_k = \underbrace{G_1 \otimes G_1 \ldots \otimes G_1}_{k \text{ раз}} = G_{k-1} \otimes G_1$$

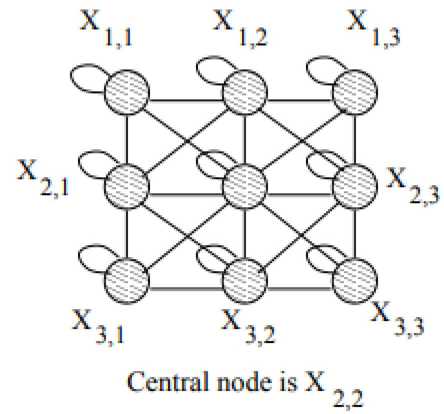
# Графы Кронекера. Пример




(a) Graph $K_1$



(b) Intermediate stage




(c) Graph $K_2 = K_1 \otimes K_1$

| | | |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 1 |

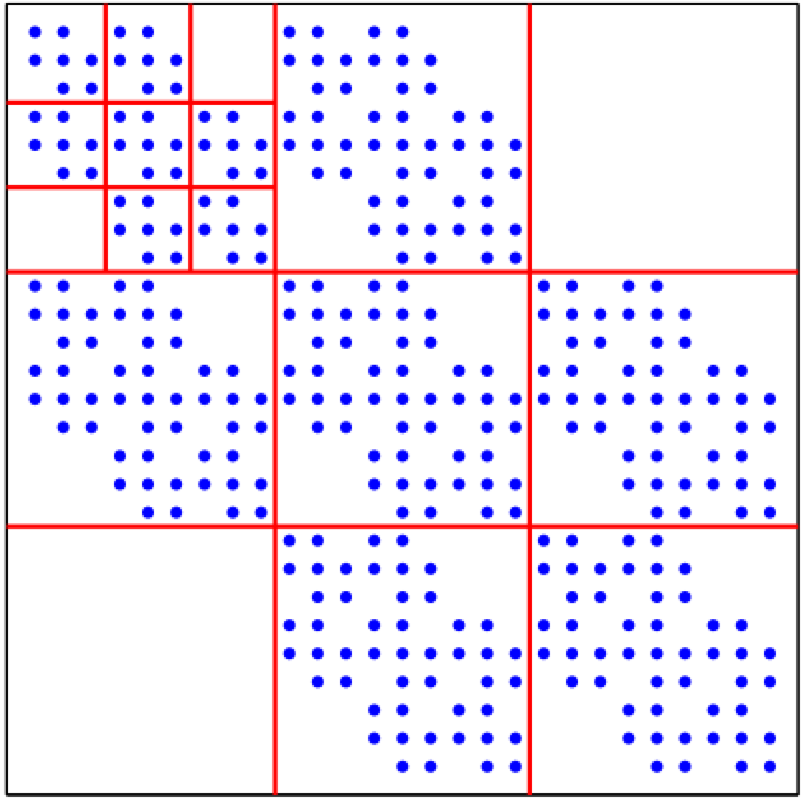
(d) Adjacency matrix of $K_1$

| | | |
|---|---|---|
| $K_1$ | $K_1$ | 0 |
| $K_1$ | $K_1$ | $K_1$ |
| 0 | $K_1$ | $K_1$ |

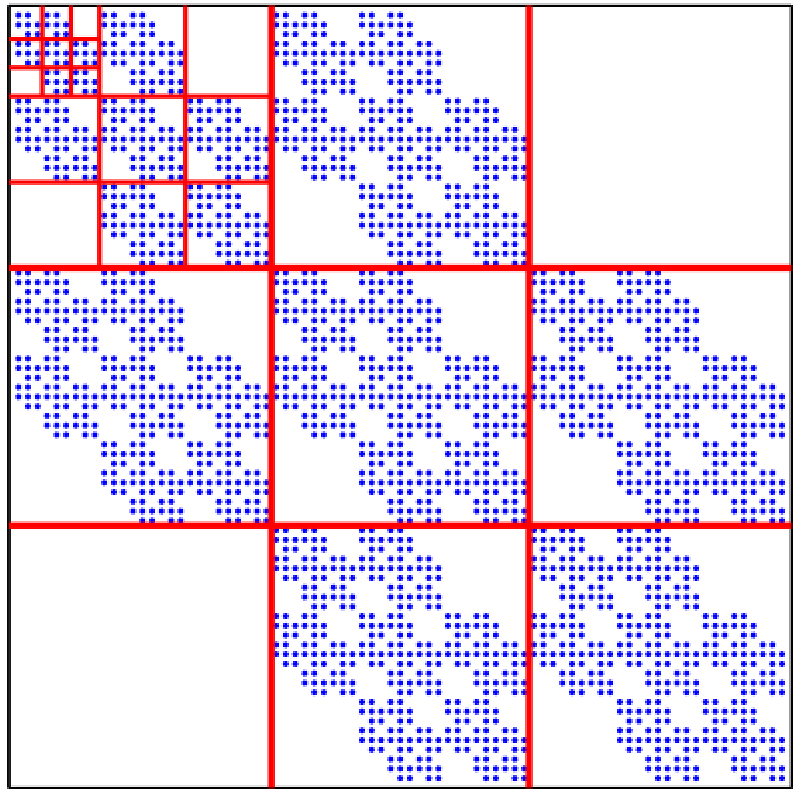
(e) Adjacency matrix of $K_2 = K_1 \otimes K_1$

Leskovec J. et al. Kronecker graphs: An approach to modeling networks // Journal of Machine Learning Research. – 2010. – Т. 11. – №. Feb. – С. 985-1042.

# Графы Кронекера. Пример



(a) $K_3$ adjacency matrix ($27 \times 27$)



(b) $K_4$ adjacency matrix ($81 \times 81$)

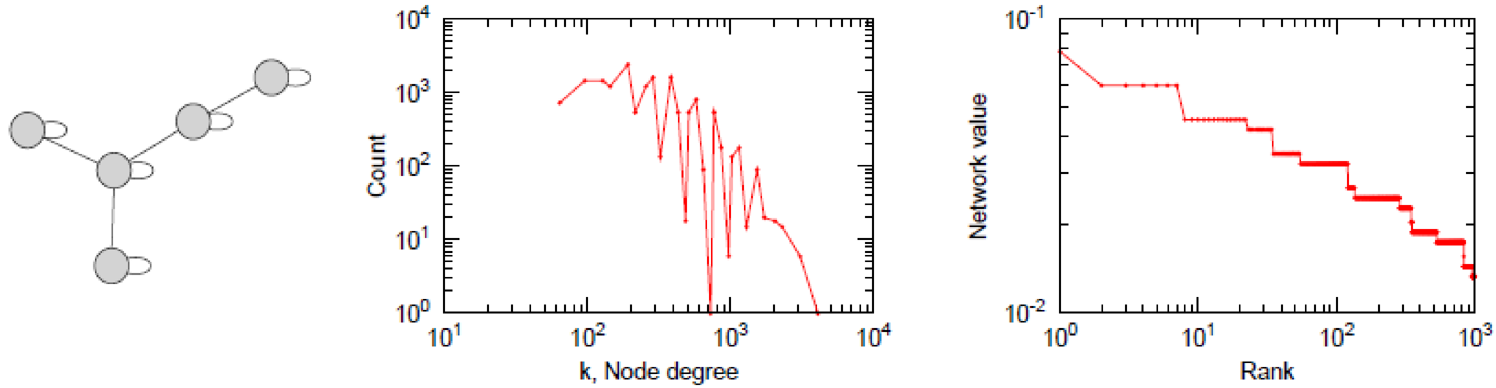
Leskovec J. et al. Kronecker graphs: An approach to modeling networks // Journal of Machine Learning Research. – 2010. – Т. 11. – №. Feb. – С. 985-1042.

# Графы Кронекера. Свойства

- Полиномиальное распределение степеней, собственных значений графа и компонент каждого собственного вектора.
- Если граф-инициатор имеет $N_1$ вершин и $E_1$ ребер, то последовательность графов Кронекера следует уплотнению степенного распределения с экспонентой $a = \log(E_1)/\log(N_1)$.
- Если граф-инициатор имеет диаметр $d$ и у каждой вершины есть петля, то граф $G_k$ имеет диаметр $d$ не зависимо от $k$. Для любого $q$ $q$-эффективный диаметр $G_k$ сходится к $d$ при росте $k$.
- Если графы G и H оба связные, но двудольные, то $G \otimes H$ несвязный, и каждая его компонента двудольная.

# Графы Кронекера. Свойства

- Недостаток модели: «ступенчатый» вид характеристик графа -




(a) Kronecker initiator $K_1$

(b) Degree distribution of $K_6$ ($6^{th}$ Kronecker power of $K_1$)

(c) Network value of $K_6$ ($6^{th}$ Kronecker power of $K_1$)

- Модификация модели – внесение вероятности появления ребер

Leskovec J. et al. Kronecker graphs: An approach to modeling networks // Journal of Machine Learning Research. – 2010. – Т. 11. – №. Feb. – С. 985-1042.

# Стохастические графы Кронекера

- Пусть $P_1$ - матрица вероятностей размера $N_1 \times N_1$. Значение $\theta_{ij} \in P_1$ обозначает вероятность того, что ребро $(i, j)$ включено в граф $G_1$. Для $k$-го произведения графа Кронекера каждое ребро $(u, v) \in G_k$ включается в граф с вероятностью $P_1^{[k]}$ .

| | |
|---|---|
| 0.6 | 0.1 |
| 0.2 | 0.4 |

$P_1$

| | | | |
|---|---|---|---|
| 0.36 | 0.06 | 0.06 | 0.01 |
| 0.12 | 0.24 | 0.02 | 0.04 |
| 0.12 | 0.02 | 0.24 | 0.04 |
| 0.04 | 0.08 | 0.08 | 0.16 |

$P_2 = P_1 \times P_1$

вероятность ребра (3,3)

Реализация графа $G_2$

# Стохастические графы Кронекера

- Стохастические графы Кронекера – это обобщение других моделей:
  – Эрдеша-Рэньи – с инициатором 1 x 1. $P_1 = [\theta_{11}] = p$
  – RMAT – с инициатором 2 x 2. В отличие от RMAT, в графе Кронекера $\sum \theta_{ij}$ кодирует также общее число графов.

# Генераторы графов

**Table 2** List of publicly available implementations sorted by name of the toolkit. Abbrv.: BA: *Barabási-Albert*, ER: *Erdős-Rényi*, ES: *Edge-Switching*, FDSM: *Fixed-Degree-Sequence-Model*, RDT: *Random Delaunay Triangulation*, RGG: *Random Geometric Graph*, RHG: *Random Hyperbolic Graph*, SBM: *Stochastic Block Model*, WS: *Watts-Strogatz*, MMod: Machine Model, SEQ: Sequential, SHM: Shared-Memory, DM: Distributed Memory, Py: Python

| Toolkit | Url & Models | Language | MMod |
|---|---|---|---|
| | Implementations of Multiple Models | | |
| GraphTool | `https://graph-tool.skewed.de` · *ES, RDT, SBM* | C++ | SHM |
| GTGraph | `http://www.cse.psu.edu/~kxm85/software/GTgraph` · *ER, R-MAT* | C | SEQ |
| IGraph | `https://igraph.org/` · *BA, ER, ES, SBM, WS* | C++, Py, R | SEQ |
| KaGen | `https://github.com/sebalamm/KaGen` · *BA, ER, RDT, RGG, RHG* | C++ | SHM, DM |
| NetworkX | `https://networkx.github.io/` · *BA*, Caveman, *ER*, Holme-Kim, *LFR, RGG, SBM, WS* | Python | SEQ |
| NetworKit | `https://networkit.github.io/` · *BA, CL*, Clustered Random Graphs, *ER, FDSM*, PubWeb, *RHG, R-MAT* | C++, Py | SHM |
| Snap | `https://snap.stanford.edu/snap` · *BA, CM*, Forest Fire, Multiplicative Attribute Graphs, *Node Copy, R-MAT* | C++ | SHM |
| | Implementations of a Single Model | | |
| Darwini | `https://issues.apache.org/jira/browse/GIRAPH-1043` · *Darwini* | Java | DM |
| FEASTPACK | `https://www.sandia.gov/~tgkolda/feastpack/` · *BTER* | MATLAB | SEQ |
| Graph500 | `https://graph500.org/` · *R-MAT* | C | DM |
| HyperGen | `https://github.com/manpen/hypergen` · *RHG* | C++ | SM |
| LFR | `https://sites.google.com/site/andrealancichinetti/files` · | C++ | SEQ |
| MUSKETEER | `https://github.com/sashagutfraind/musketeer` · planar version: `https://github.com/isafro/Planar-MUSKETEER` | Python | SEQ |
| R-MAT | `https://github.com/lorenzhs/rmat` · *R-MAT* | C++ | SHM |

Penschuck M. et al. Recent Advances in Scalable Network Generation //arXiv preprint arXiv:2003.00736. – 2020.

# Генераторы графов

- RMAT графы https://github.com/lorenzhs/rmat
- Графы Кронекера https://github.com/graph500/graph500
- SSCA-2, Erdös-Rényi, R-MAT https://github.com/Bader-Research/GTgraph
- R-MAT https://github.com/farkhor/PaRMAT
- https://github.com/KarlsruheGraphGeneration/KaGen

# Литература по случайным графам

- Математическое описание моделей:
  1. Колчин В.Ф. Случайные графы. – М.: Физматлит, 2004. – 256 с.
  2. Райгородский А. Модели случайных графов. – М.: МЦНМО, 2017.
  3. Видеолекции А. Райгородского для Школы Анализа Данных Yandex, на Coursera https://ru.coursera.org/learn/sluchajnye-graphy.
- Алгоритмическое описание моделей и свойств сетей:
  1. Newman M. Networks. – Oxford university press, 2018.
  2. Chakrabarti D., Faloutsos C. Graph mining: laws, tools, and case studies //Synthesis Lectures on Data Mining and Knowledge Discovery. – 2012. – Т. 7. – №. 1. – С. 1-207.

# Заключение

- Данные, представленные в виде графа, используются во многих прикладных областях. Доступны коллекции реальных графов большого порядка
- Модель графа используется, если реальных данных недостаточно или вычисления на реальном графе затратно по времени. Также модельные графы используются для тестирования алгоритмов.
- Графы больших сетей имеют некоторые специальные свойства (степени вершин распределены согласно степенной функции, малый диаметр, справедлив степенной закон уплотнения и др.). Для них построены специальные модели

# Заключение

- Первая модель случайного графа – модель Эрдеша–Рэньи. В ней для графа с $n$ вершинами появление каждого из всех возможных $C_n^2$ ребер равновероятно
- Ряд моделей реализует принцип предпочтительного соединения: наращивание графа вокруг некоторого ядра, учитывая степени вершин. Например, модель Барабаши – Альберт.
- При реализации модели важно корректно генерировать выборку из множества ребер
- В бенчмарках часто используются графы RMAT и графы Кронекера. Они моделируют графы со степенным распределением вершин (например, графы Интернета и социальных сетей)

# Литература


1. Erdős P., Rényi A. On the evolution of random graphs // Publ. Math. Inst. Hung. Acad. Sci. – 1960. – Т. 5. – №. 1. – С. 17-60.
2. Barabasi L.-A., Albert R. Emergence of scaling in random networks // Science. 1999. V. 286. P. 509-512.
3. Chakrabarti D., Zhan Y., Faloutsos C. R-MAT: A recursive model for graph mining //Proceedings of the 2004 SIAM International Conference on Data Mining. – Society for Industrial and Applied Mathematics, 2004. – С. 442-446.
4. Leskovec J. et al. Kronecker graphs: An approach to modeling networks //Journal of Machine Learning Research. – 2010. – Т. 11. – №. Feb. – С. 985-1042.
5. Penschuck M. et al. Recent Advances in Scalable Network Generation //arXiv preprint arXiv:2003.00736. – 2020.
6. Batagelj V., Brandes U. Efficient generation of large random networks //Physical Review E. – 2005. – Т. 71. – №. 3. – С. 036113.
7. Райгородский А. М. Модели случайных графов и их применения //Труды Московского физико-технического института. – 2010. – Т. 2. – №. 4.

# Контакты

Нижегородский государственный университет

http://www.unn.ru

Институт информационных технологий, математики и механики

http://www.itmm.unn.ru

Пирова А.Ю.

anna.pirova@itmm.unn.ru