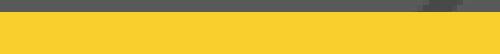


Граф содержит Гамильтонов цикл: $0 \Rightarrow 1 \Rightarrow 9 \Rightarrow 10 \Rightarrow 11 \Rightarrow 3 \Rightarrow 5 \Rightarrow 2 \Rightarrow 4 \Rightarrow 12 \Rightarrow 8 \Rightarrow 7 \Rightarrow 0$

«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

ЗАЩИТА КУРСОВОЙ РАБОТЫ ПО ТЕМЕ: ГАМИЛЬТОНОВЫ ЦИКЛЫ



4 2018

Выполнили студенты группы О9-
732: Хасаншин Нияз, Гайфуллина
Карина, Ирина Овчинникова и
Коробейников Кирилл

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

-
- 1.1 Основные определения и результаты
 - 1.2 Теоремы достаточности гамильтонова графа
 - 2. Методы отыскания гамильтоновых циклов
 - 3. Реализация методов отыскания гамильтоновых циклов
 - 3.1 Алгебраические методы
 - 3.2 Метод перебора Робертса и Флореса
 - 3.2.1 Улучшение метода Робертса и Флореса
 - 3.3 Рекурсивный метод
 - 3.4 Метод ветвей и границ
 - Заключение
 - Список литературы
 - Приложение



ЦЕЛЬ КУРСОВОЙ РАБОТЫ

**Поиск метода и создание программы
оптимального нахождения
гамильтоновых циклов в графах**



ИСТОРИЯ

В 1859 г. сэр Вильям Гамильтон, известный ирландский математик, написавший теорию комплексного числа и кватерниона, создал детскую игру-головоломку, в которой нужно было совершить «кругосветное путешествие» по 20 городам, расположенным в различных частях земного шара. Все города соединялись дорогами с тремя городами по соседству так, что дорожная сеть образовывала 30 ребер додекаэдра, в вершинах которого находились города.

Обязательным условием было требование посетить каждый город, за исключением первого, лишь один раз.

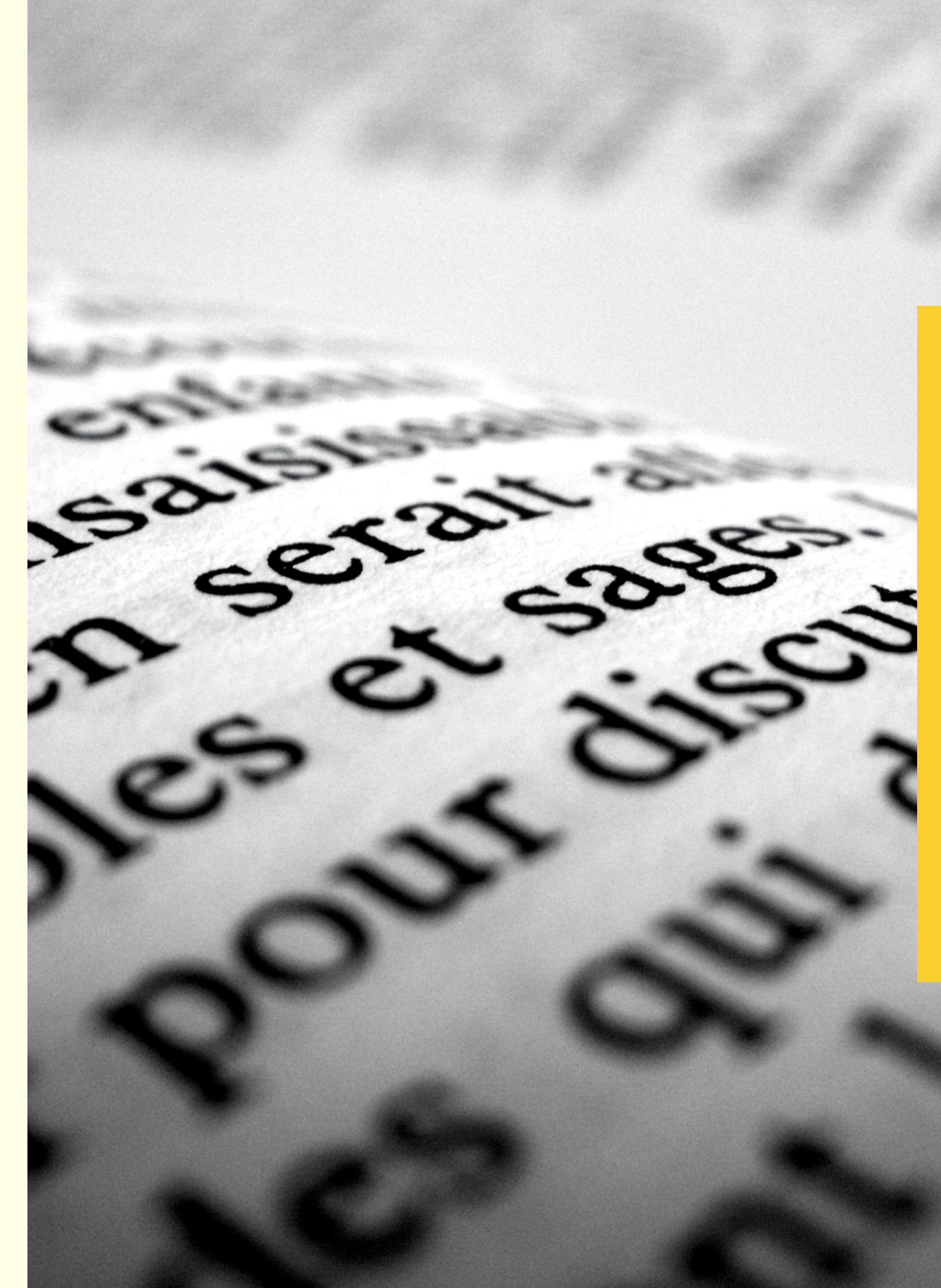
ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Гамильтоновым путём (англ. **Hamiltonian path**) называется простой путь, по которому проходит через каждую вершину графа совершается только один раз.

Гамильтоновым циклом (англ. **Hamiltonian cycle**) называют замкнутый гамильтонов путь.

Граф называется полугамильтоновым (англ. **Semihamiltonian graph**), если он содержит гамильтонов путь.

Граф называется гамильтоновым (англ. **Hamiltonian graph**), если он содержит гамильтонов цикл



КАКИЕ УСЛОВИЯ НЕОБХОДИМЫ ДЛЯ СУЩЕСТВОВАНИЯ ГАМИЛЬТОНОВЫХ ЦИКЛОВ ?

ТЕОРЕМА ДИРАКА

Если $n \geq 3$ и $\deg v \geq n/2$ для любой вершины v неориентированного графа G , то G — гамильтонов граф.

ТЕОРЕМА ОРЕ

Если $n \geq 3$ и $\deg u + \deg v \geq n$ для любых двух различных несмежных вершин u и v неориентированного графа G , то G — гамильтонов граф

ТЕОРЕМА ГУЙЯ-УРИ

Пусть G — сильносвязный ориентированный граф.
 $\deg^+v \geq n/2$;
 $\deg^-v \geq n/2 \Rightarrow G$ — гамильтонов

ТЕОРЕМА ХВАТАЛА

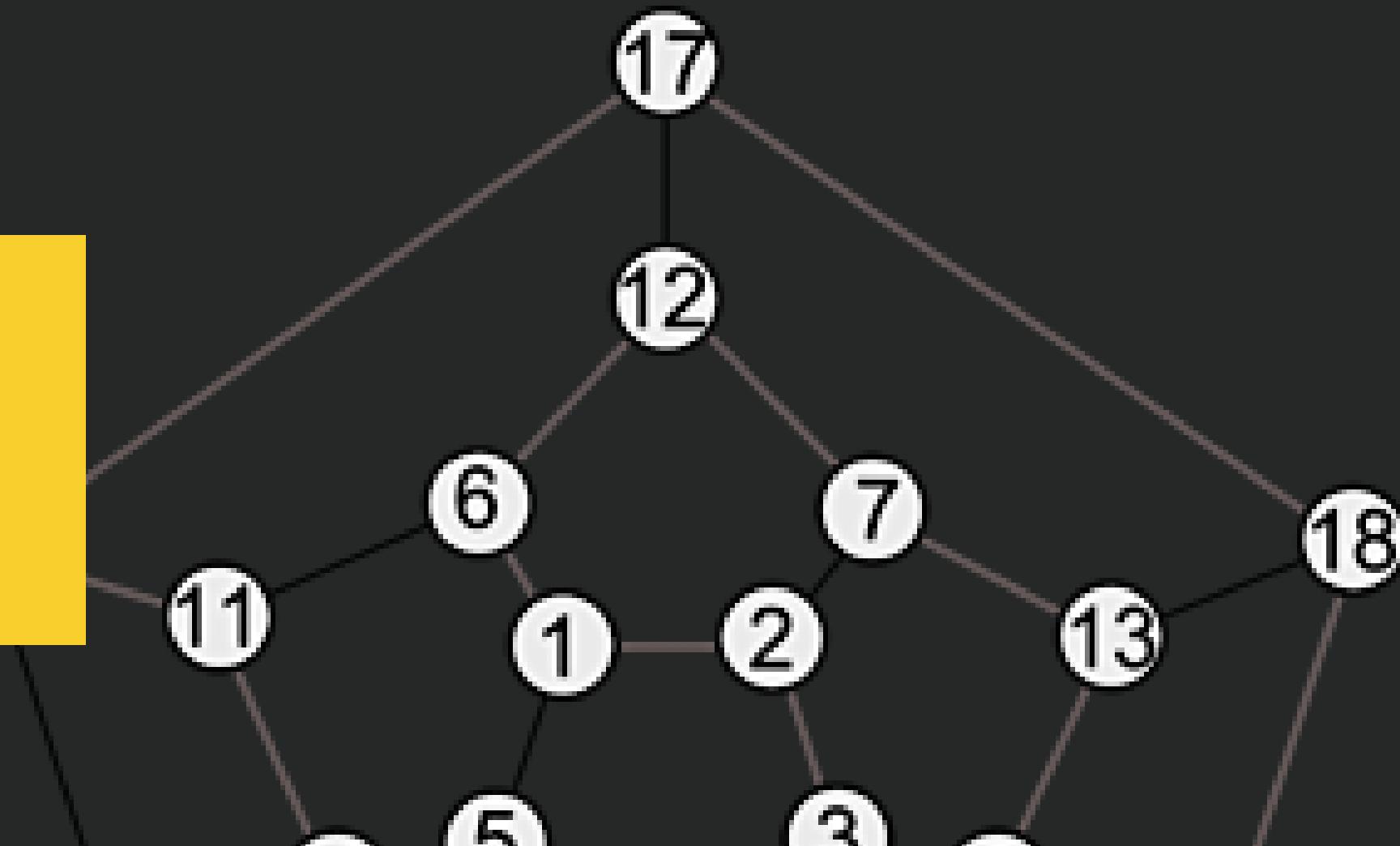
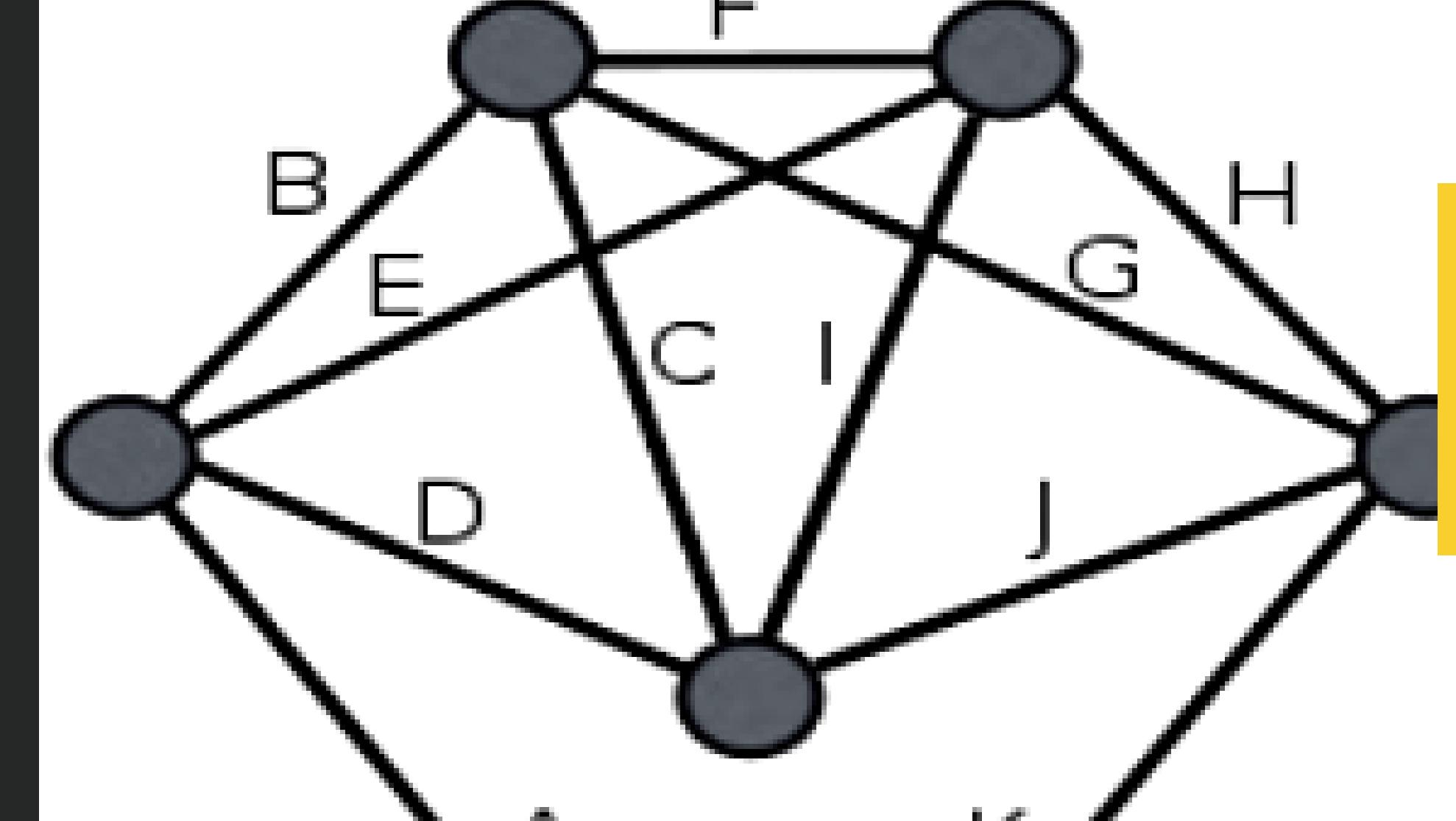
Пусть:

- * G – связный граф,
- * $n = |V_G| \geq 3$ – количество вершин,
- * $d_1 \leq d_2 \leq \dots \leq d_n$ – его последовательность степеней.

Тогда если $\forall k \in \mathbb{N}$ верна импликация:

$$d_k \leq k < n/2 \Rightarrow d_{n-k} \geq n-k,$$

то граф G гамильтонов



ТЕОРЕМА ПОША

Пусть граф G имеет $n \geq 3$ вершин и выполнены следующие два условия:

- * для всякого $k, 1 \leq k < (n-1)/2$, число вершин со степенями, не превосходящими k , меньше чем k ;

- * для нечетного n число вершин степени $(n-1)/2$ не превосходит $(n-1)/2$, тогда G – гамильтонов граф.

ОСНОВНЫЕ ЗАДАЧИ



ЗАДАЧА О КОММИВОЯЖЕРЕ

Коммивояжер должен посетить N городов, побывать в каждом из них ровно по одному разу и завершить свой путь в том городе, с которого он начал. В какой последовательности ему нужно обойти города, чтобы общая длина его пути была самой короткой?

ЗАДАЧА О ШАХМАТНОМ КОНЕ

Возможно ли, начиная с произвольной клетки шахматной доски, обойти конем последовательно все 64 поля по одному разу и вернуться в исходное?

ЗАДАЧА ПРО БАНКЕТ

Компанию из нескольких человек требуется рассадить за круглым столом таким образом, чтобы по обе стороны от каждого сидели люди, которых он знает. Очевидно, для решения этой задачи нужно найти гамильтонов цикл в графе знакомств компании

МЕТОДЫ ОТЫСКАНИЯ ГАМИЛЬТОНОВЫХ ЦИКЛОВ

АЛГЕБРАИЧЕСКИЕ
МЕТОДЫ

МЕТОД ПЕРЕБОРА
РОБЕРТСА И ФЛОРЕСА

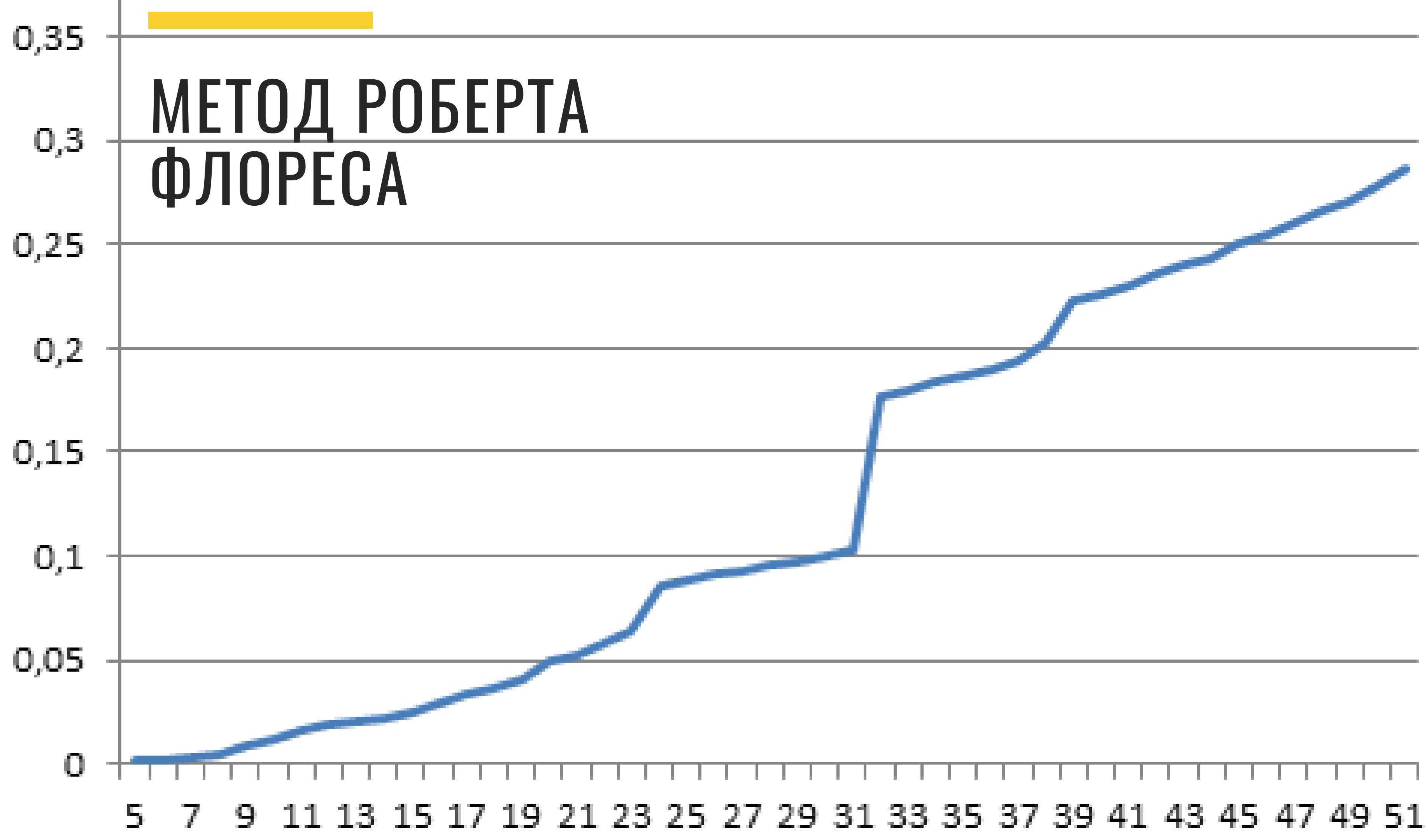
АЛГОРИТМЫ С
ВОЗВРАТОМ

МЕТОД ВЕТВЕЙ И
ГРАНИЦ

АЛГЕБРАИЧЕСКИЙ МЕТОД



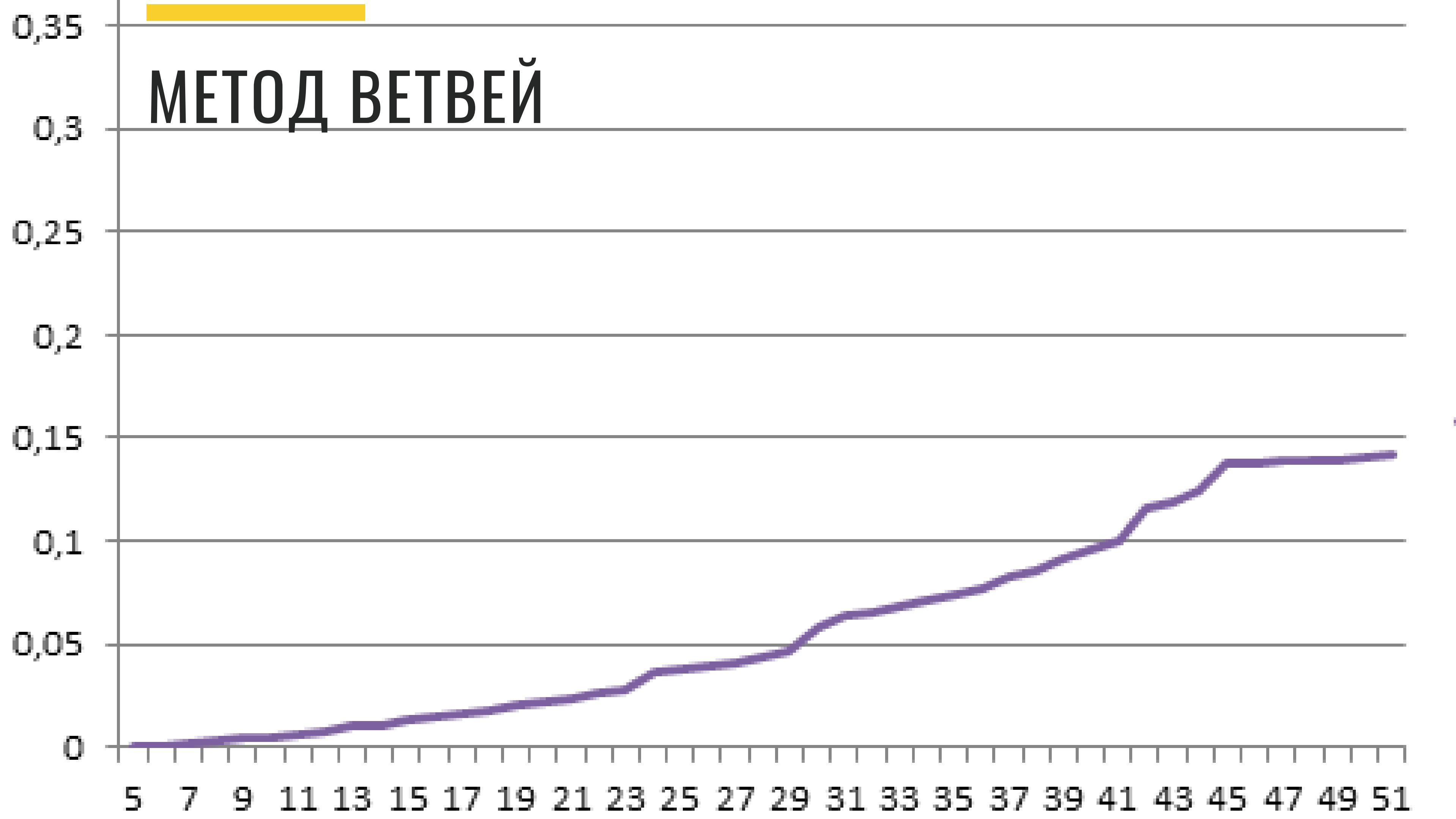
**МЕТОД РОБЕРТА
ФЛОРЕСА**

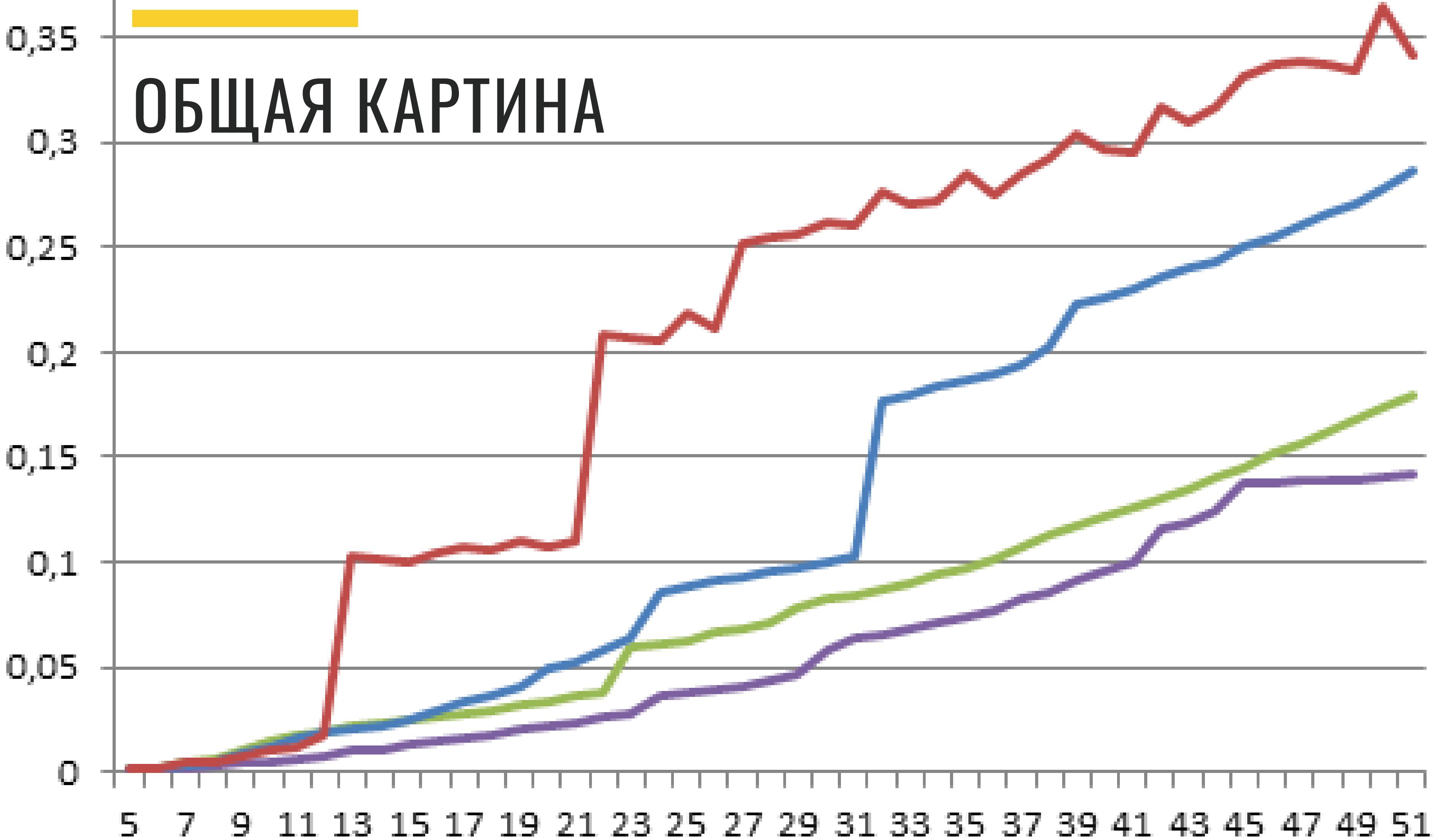


РЕКУРСИВНЫЙ ПОДХОД



МЕТОД ВЕТВЕЙ





View Navigate Code Refactor Run Tools VCS Window Help



main

```

7 import numpy as np
8 import os
9 import math
10
11 #print("Введите данные своего графа, если данные с файла то введите 1, если данные будут введены с руки введите 2, либо 3 если данные заданы в программе")
12 ras = int(input())
13 if (ras == 1):
14     f = open('Data', 'r')
15     data = f.read()
16     #print(data)
17 elif (ras == 2):
18     print("Введите данные...")
19     #data = input()
20 elif ((ras != 1) and (ras != 2)):
21     #print ("Данные заданы в программе")
22
23 def doit(data):
24     print("Введите число вершин")
25     number = int(input())
26     start_time = time.time()
27
28     def main():
29         i = 0
30
31         while i < number:
32             i = i + 1
33             if i > 3:
34                 timeforprog = (time.time() - start_time)
35                 adj = np.random.randint(0, 2, (i, i))
36                 #a_random_directed_graph
37                 G = nx.from_numpy_matrix(adj) # generator
38                 nx.draw(G) #рисунок
39                 plt.show()
40                 print("Nodes of graph: ")
41                 print(G.nodes())
42                 print("Edges of graph: ")
43                 print(G.edges())
44                 A = adj
45                 n = len(G.nodes)
46                 print(n)
47                 Visited = [False] * n
48                 Path = []
49
50                 def hamilton(curr):
51                     Path.append(curr)
52
53                     if curr == n-1:
54                         print("Path found: ", Path)
55                         return True
56
57                     for i in range(n):
58                         if adj[curr][i] == 1 and Visited[i] == False:
59                             Visited[i] = True
60                             if hamilton(i):
61                                 return True
62                             Visited[i] = False
63
64                 main() > while i < number > if i > 3

```

Running console x main x main(2) x

1162805557250977 seconds ---

f graph:

2, 3, 4]

f graph:

, (0, 2), (0, 3), (1, 3), (1, 2), (2, 2), (2, 4), (3, 4), (4, 4)]

5395259857177734 seconds ---

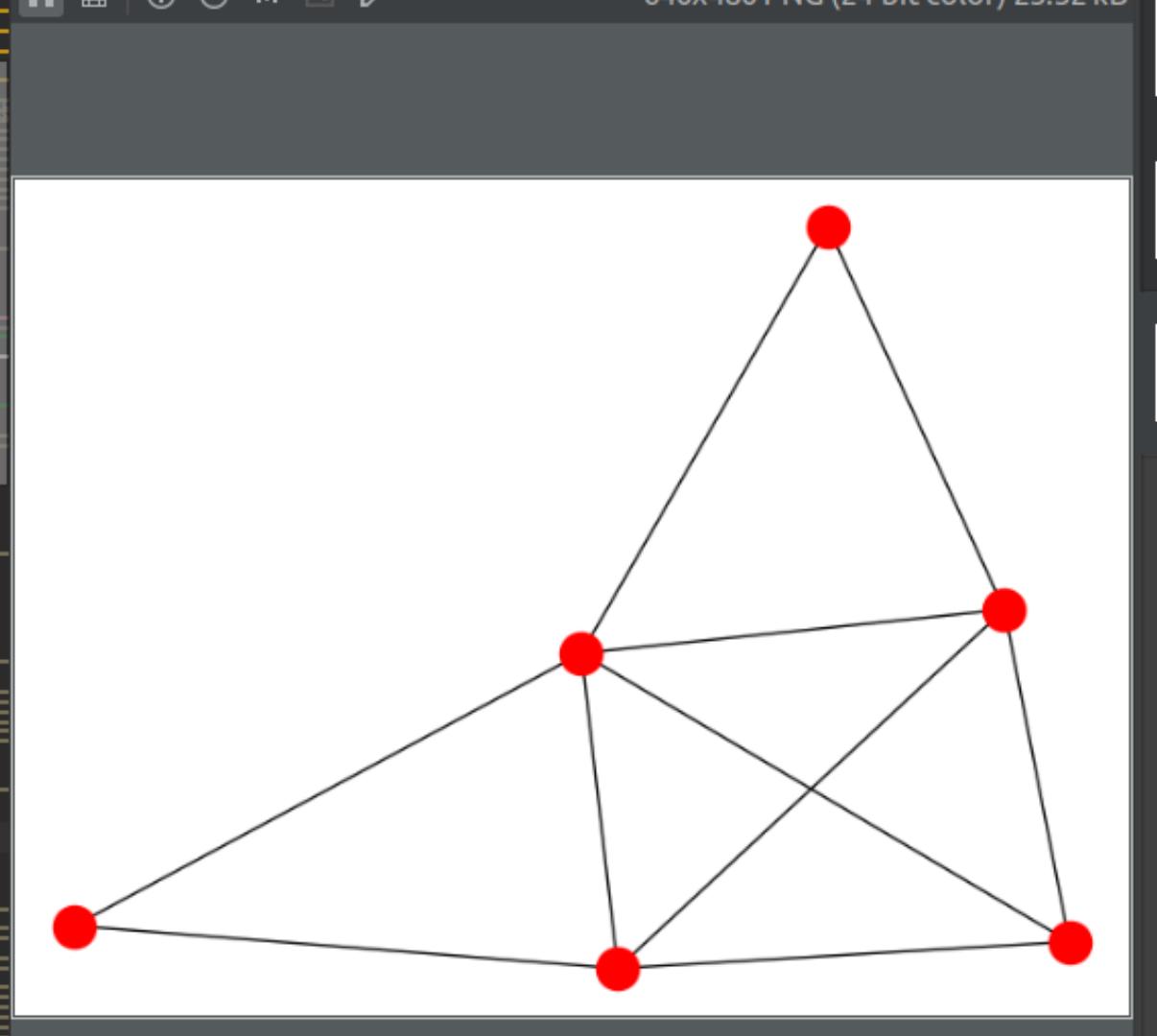
f graph:

2, 3, 4, 5]

SciView: Data Plots



640x480 PNG (24-bit color) 23.32 kB



Documentation: adj x

adj: Union[int, ndarray[int]] = np.random.randint(0, 2, (i, i))

► Special Variables

number = {int} 6

start_time = {float} 1545347333.304797

ЗАКЛЮЧЕНИЕ

В данной курсовой работе мы познакомились с основными понятиями и определениями, связанными с гамильтоновыми графами и циклами. Досконально изучили методы их отыскания. Так же мы рассмотрели ту часть теории сложности, которая затрагивает задачи отыскания гамильтонова цикла. Провели тестирование методов на графах различной сложности и величины, выявили свои минусы и плюсы каждого из методов, а также опытным путем обнаружили и зафиксировали зависимость времени выполнения алгоритма от количества вершин в графе. На основе этих данных, нам удалось построить график сравнения времени выполнения всех доступных во время исследования алгоритмов. И в итоге, данные проведенных исследований позволили нам написать оптимальную и функционирующую программу отыскания гамильтонова цикла в графе.

СПАСИБО ЗА ВНИМАНИЕ !!!