

Фортов Егор, БПИ214

КРАТКО ПРО СЛОЖНОСТИ АНАЛИЗИРУЕМЫХ АЛГОРИТМОВ:

- Алгоритм Дейкстры - $O(E \log V)$, где E - количество ребер в графе, V - количество вершин.
- Алгоритм Флойда-Уоршелла - $O(V^3)$, где V - количество вершин в графе.
- Алгоритм Беллмана-Форда - $O(E \cdot V)$, где E - количество ребер в графе, V - количество вершин.

Графики зависимости отдельно по каждому типу графа

ПОЛНЫЕ ГРАФЫ

Примечание: в полном графе из n вершин будет

$$\frac{n \cdot (n - 1)}{2}$$

ребер.

С учетом этого в случае полных графов (вершины):

- Сложность Дейкстры:
 $O(n^2 \log n)$
- Сложность Флойда-Уоршелла:
 $O(n^3)$
- Сложность Беллмана-Форда:
 $O(n^3)$

В случае полных графов (ребра):

- Сложность Дейкстры:
 $O(E \log E)$
- Сложность Флойда-Уоршелла:
 $O(E^{1.5})$
- Сложность Беллмана-Форда:
 $O(E^{1.5})$

```
import pandas as pd
import matplotlib.pyplot as plt
```

```

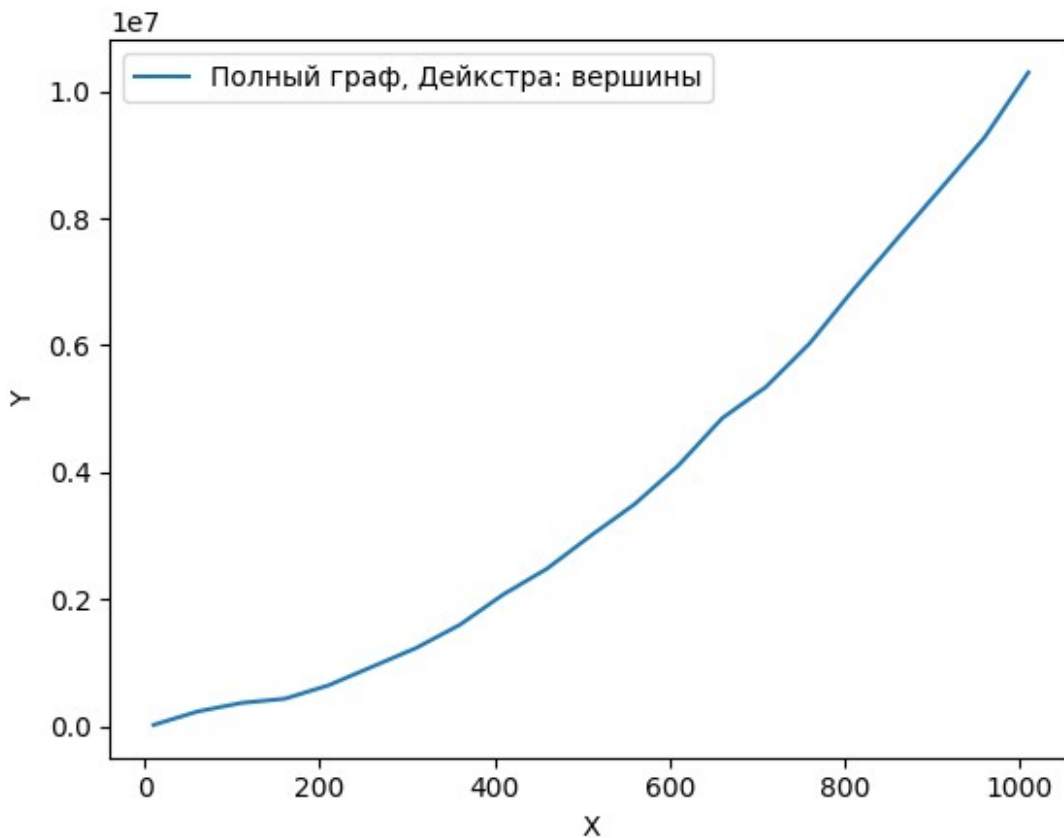
# чтение csv файла
df = pd.read_csv('1_vertexes_dijkstra.csv', sep=',')

# создание графика
plt.plot(df['x'], df['y'], label='Полный граф, Дейкстра: вершины')

# настройка осей и легенды
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()

# отображение графика
plt.show()

```



```

import pandas as pd
import matplotlib.pyplot as plt

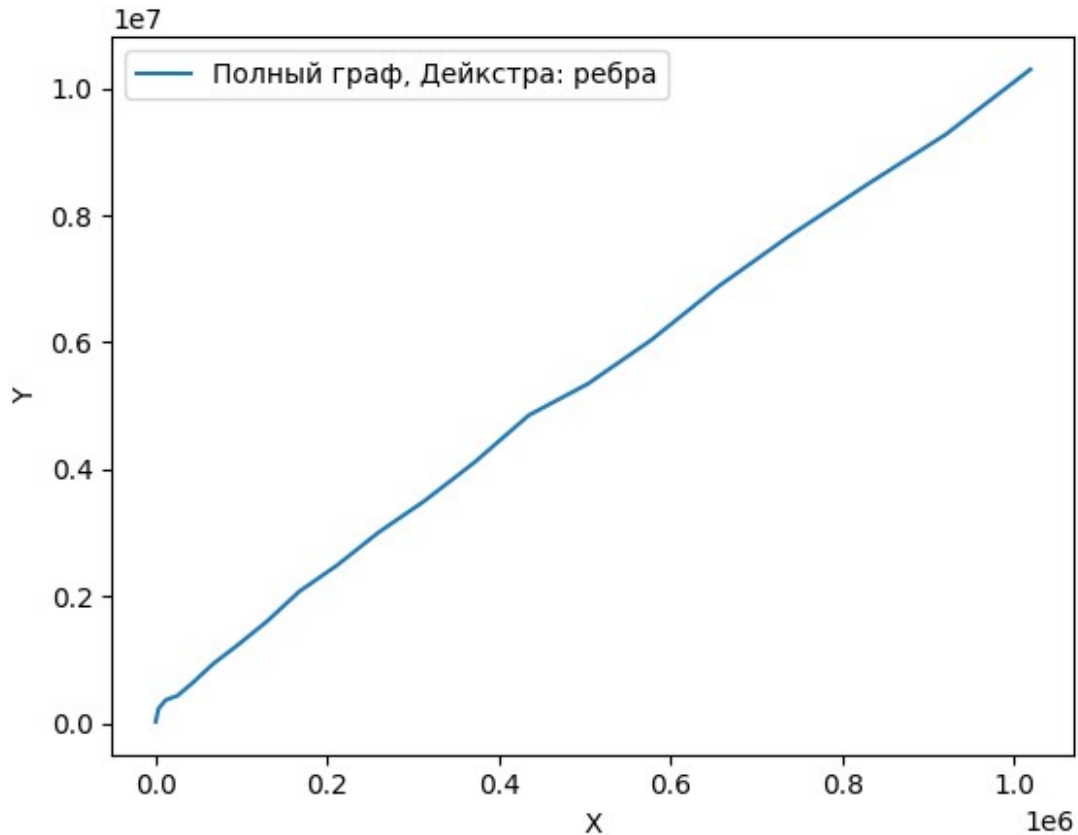
# чтение csv файла
df = pd.read_csv('1_edges_dijkstra.csv', sep=',')

# создание графика
plt.plot(df['x'], df['y'], label='Полный граф, Дейкстра: ребра')

```

```
# настройка осей и легенды
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()
```

```
# отображение графика
plt.show()
```



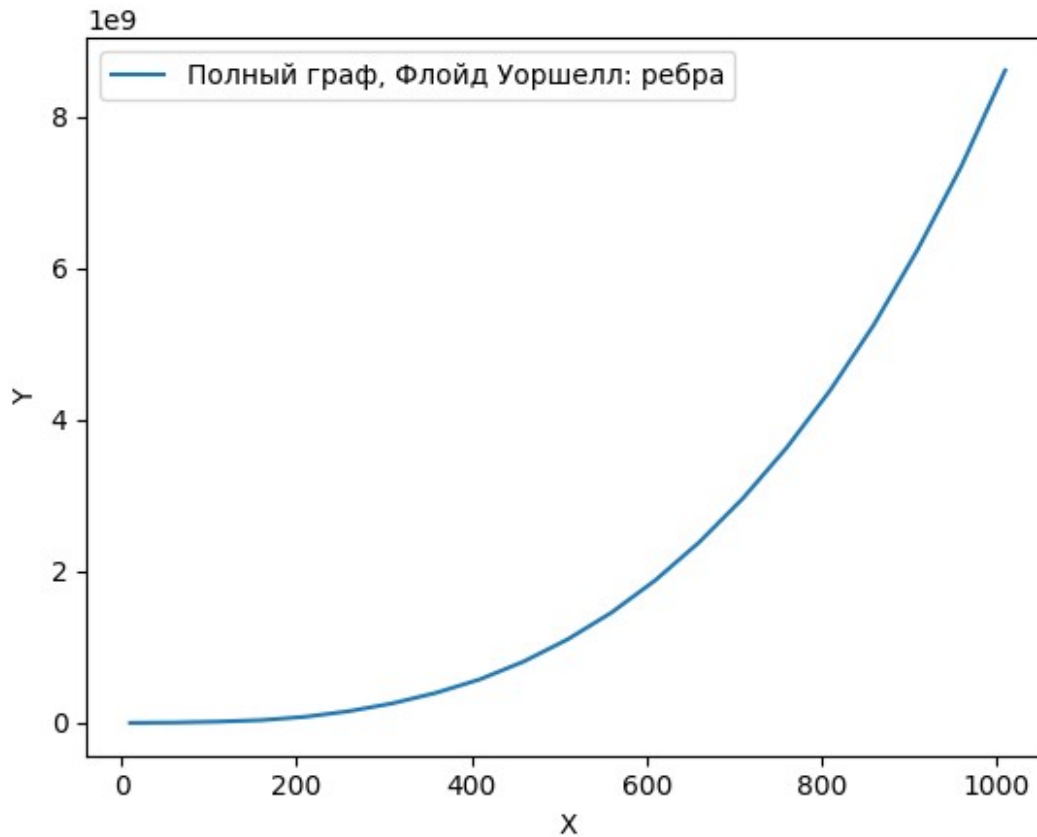
```
import pandas as pd
import matplotlib.pyplot as plt
```

```
# чтение csv файла
df = pd.read_csv('1_vertexes_floyd_warshall.csv', sep=',')
```

```
# создание графика
plt.plot(df['x'], df['y'], label='Полный граф, Флойд Уоршелл:
вершины')
```

```
# настройка осей и легенды
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()
```

```
# отображение графика
plt.show()
```



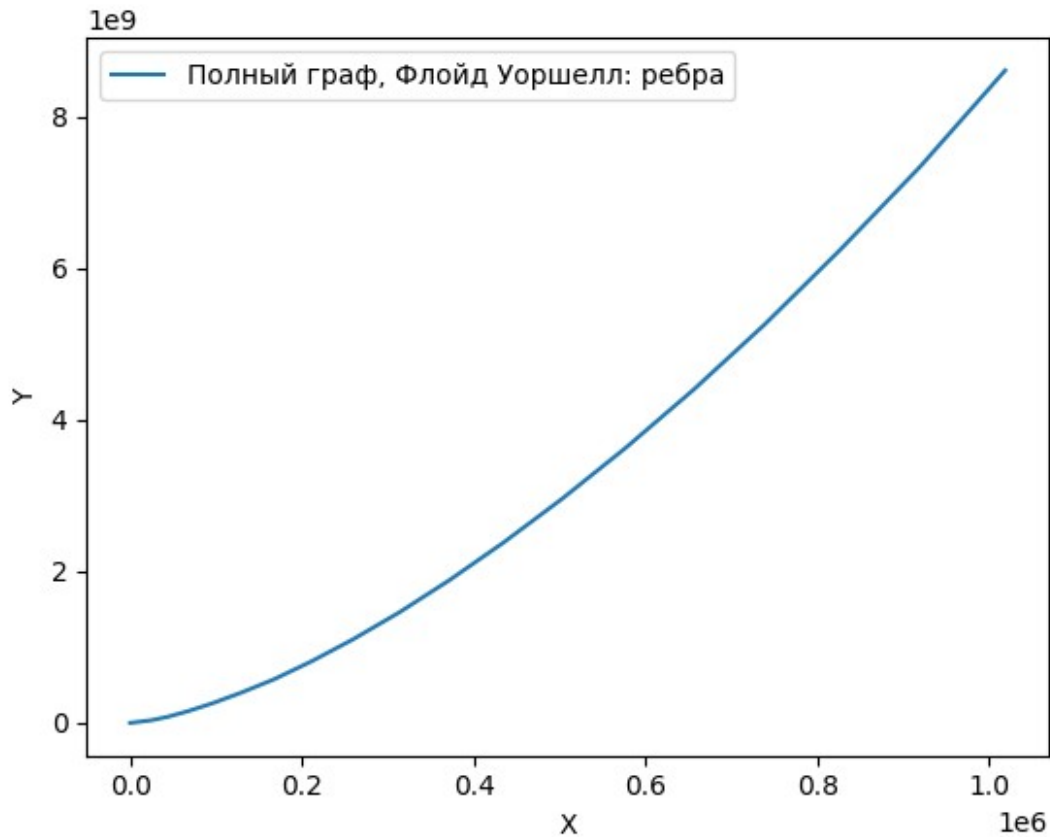
```
import pandas as pd
import matplotlib.pyplot as plt

# чтение csv файла
df = pd.read_csv('1_edges_floyd_warshall.csv', sep=',')

# создание графика
plt.plot(df['x'], df['y'], label='Полный граф, Флойд Уоршелл: ребра')

# настройка осей и легенды
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()

# отображение графика
plt.show()
```



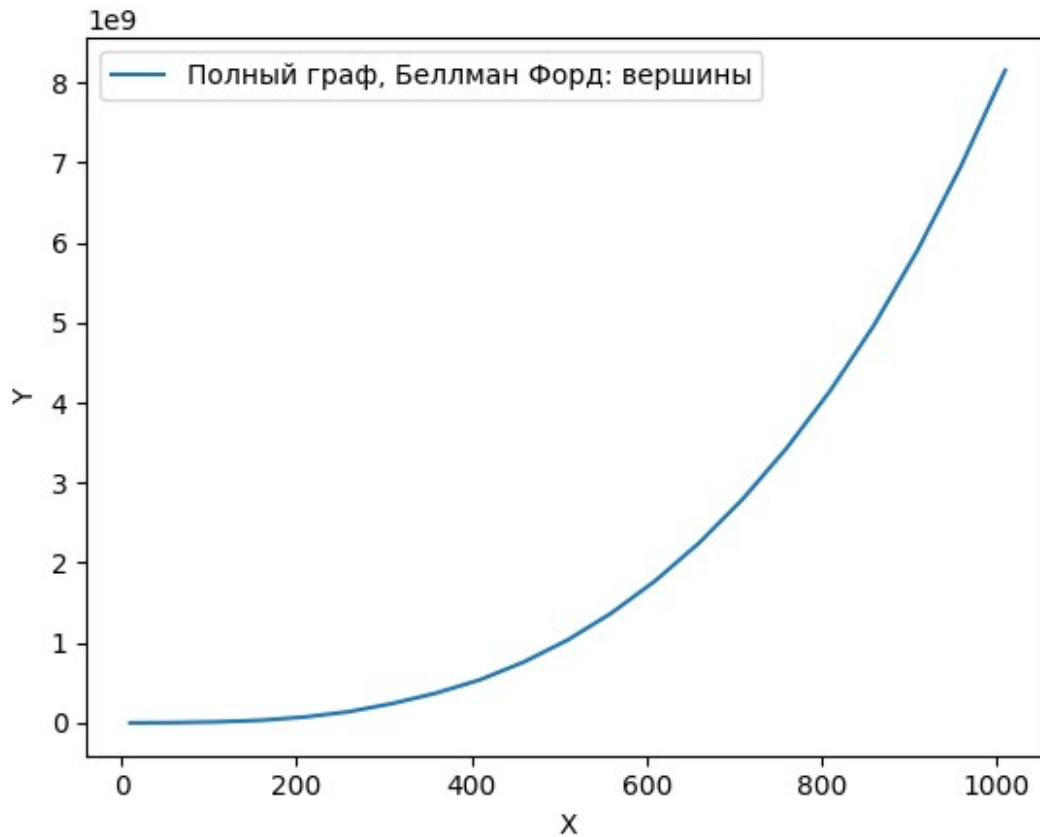
```
import pandas as pd
import matplotlib.pyplot as plt

# чтение csv файла
df = pd.read_csv('1_vertexes_bellman_ford.csv', sep=',')

# создание графика
plt.plot(df['x'], df['y'], label='Полный граф, Беллман Форд: вершины')

# настройка осей и легенды
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()

# отображение графика
plt.show()
```



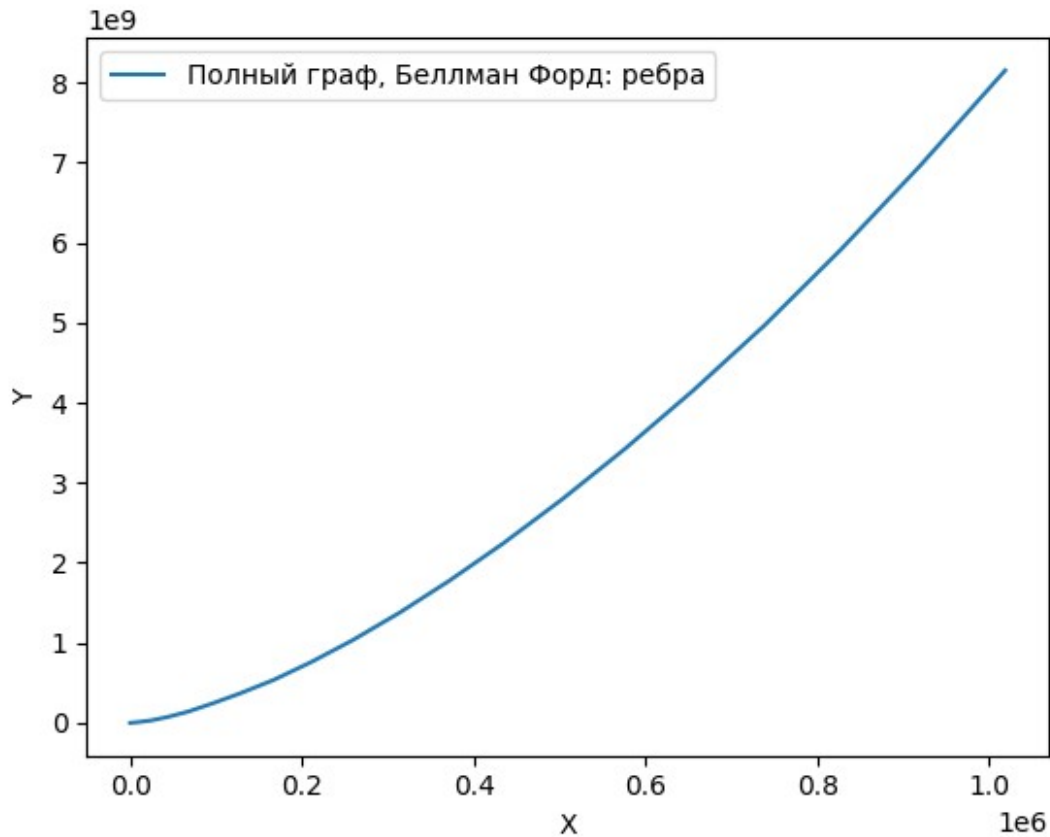
```
import pandas as pd
import matplotlib.pyplot as plt

# чтение csv файла
df = pd.read_csv('1_edges_bellman_ford.csv', sep=',')

# создание графика
plt.plot(df['x'], df['y'], label='Полный граф, Беллман Форд: ребра')

# настройка осей и легенды
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()

# отображение графика
plt.show()
```



СВЯЗНЫЕ ГРАФЫ

Примечание: Для связного графа с n вершинами и коэффициентом плотности d количество ребер E может быть вычислено как

$$E = (n * (n - 1) / 2) * d$$

Т.е. в нашем случае

$$E = n * (n - 1)$$

С т.з. амортизац. сложности - все так же, как с полными графами (отличается только амортизационный коэффициент)

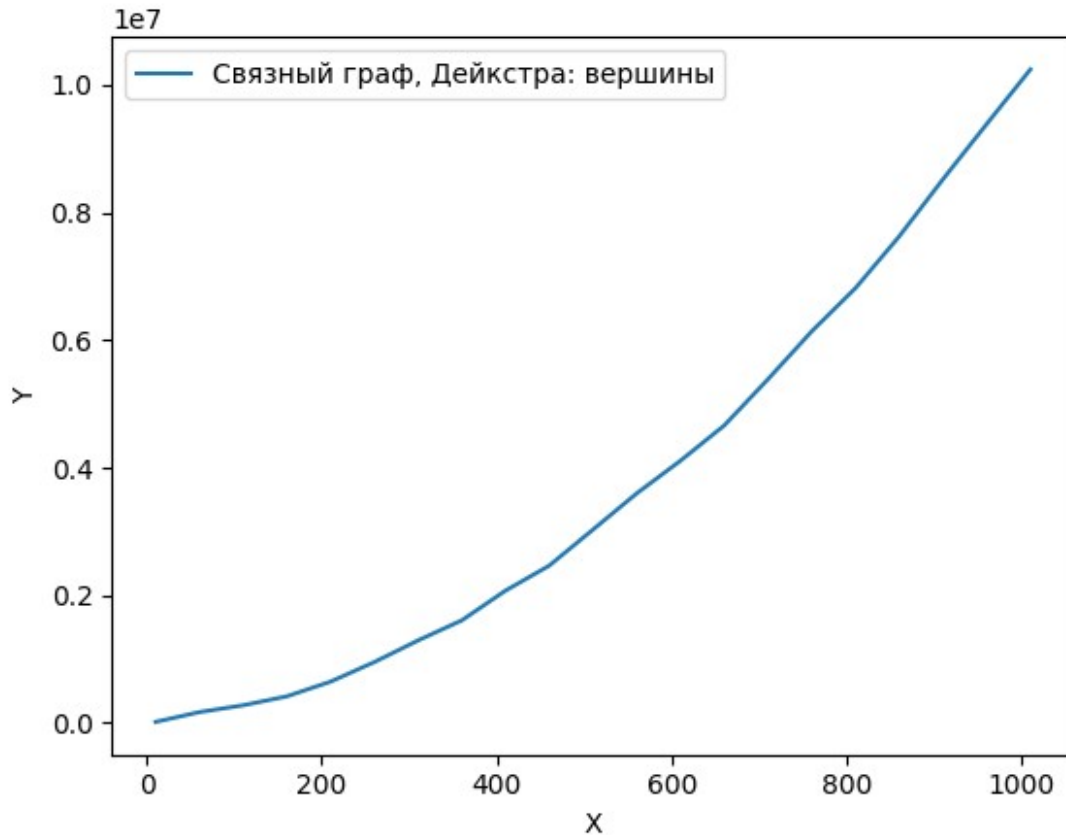
```
import pandas as pd
import matplotlib.pyplot as plt

# чтение csv файла
df = pd.read_csv('2_vertexes_dijkstra.csv', sep=',')

# создание графика
plt.plot(df['x'], df['y'], label='Связный граф, Дейкстра: вершины')
```

```
# настройка осей и легенды
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()
```

```
# отображение графика
plt.show()
```



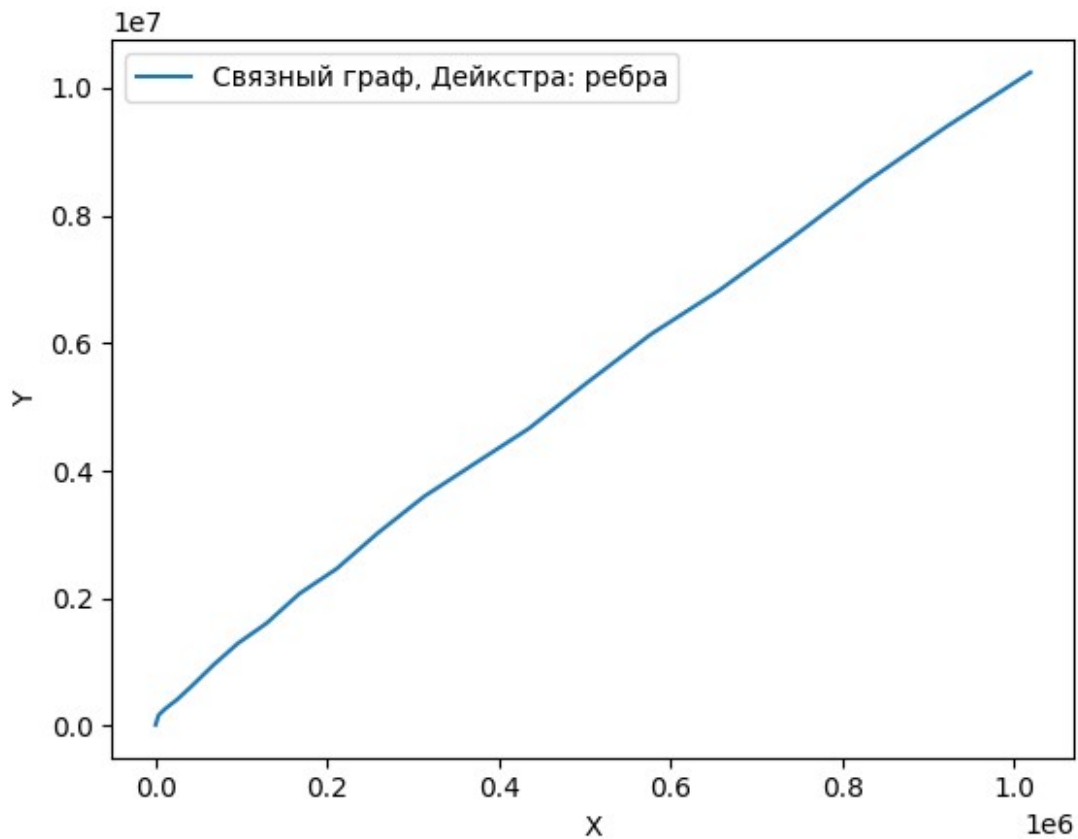
```
import pandas as pd
import matplotlib.pyplot as plt
```

```
# чтение csv файла
df = pd.read_csv('2_edges_dijkstra.csv', sep=',')
```

```
# создание графика
plt.plot(df['x'], df['y'], label='Связный граф, Дейкстра: ребра')
```

```
# настройка осей и легенды
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()
```

```
# отображение графика
plt.show()
```

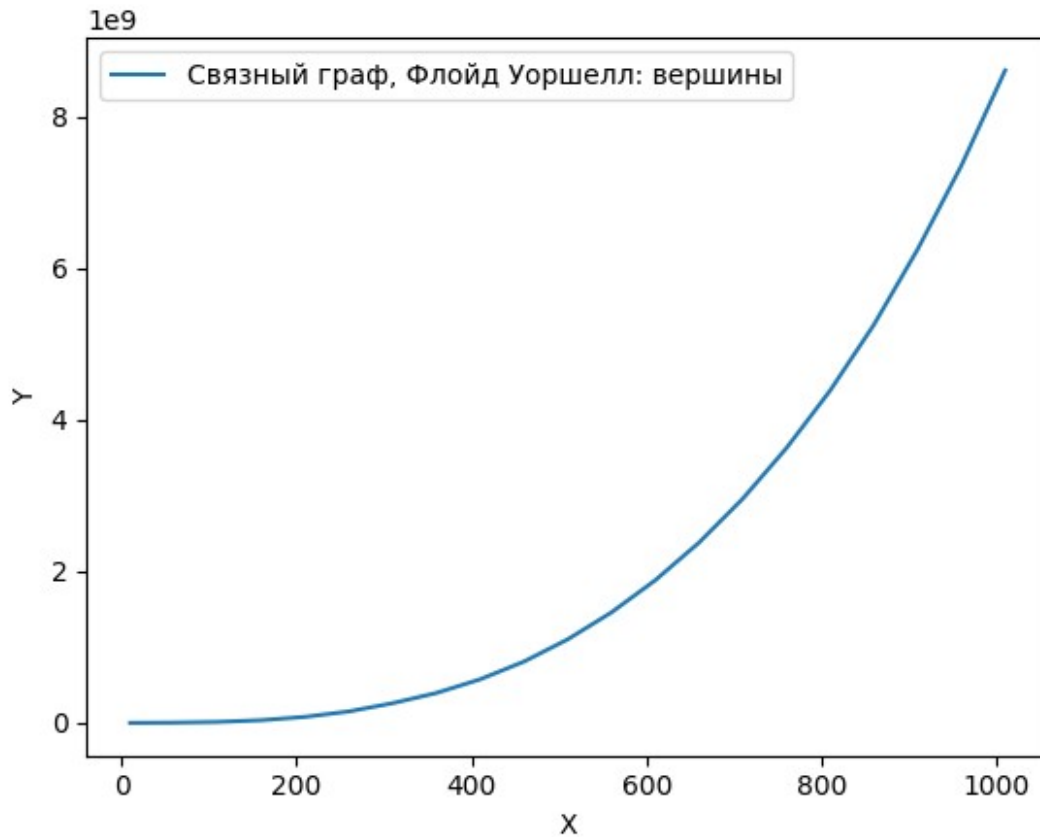
```
import pandas as pd
import matplotlib.pyplot as plt

# чтение csv файла
df = pd.read_csv('2_vertexes_floyd_warshall.csv', sep=',')

# создание графика
plt.plot(df['x'], df['y'], label='Связный граф, Флойд Уоршелл:
вершины')

# настройка осей и легенды
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()

# отображение графика
plt.show()
```



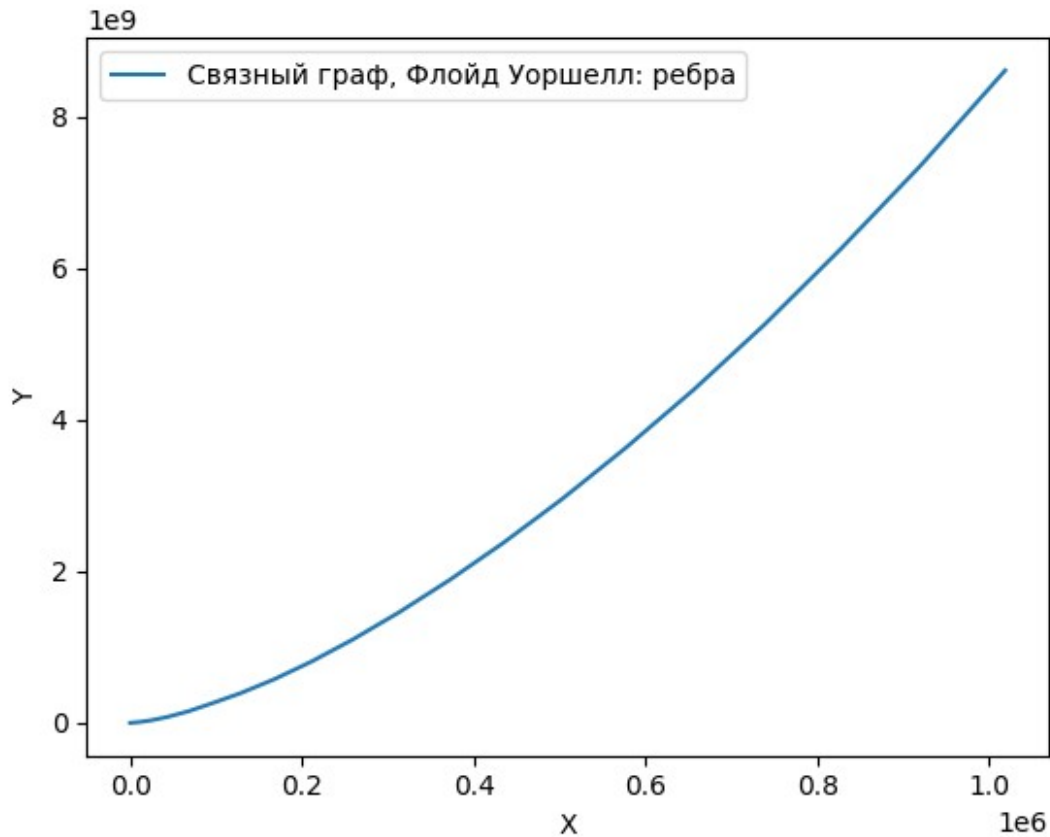
```
import pandas as pd
import matplotlib.pyplot as plt

# чтение csv файла
df = pd.read_csv('2_edges_floyd_warshall.csv', sep=',')

# создание графика
plt.plot(df['x'], df['y'], label='Связный граф, Флойд Уоршелл: ребра')

# настройка осей и легенды
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()

# отображение графика
plt.show()
```



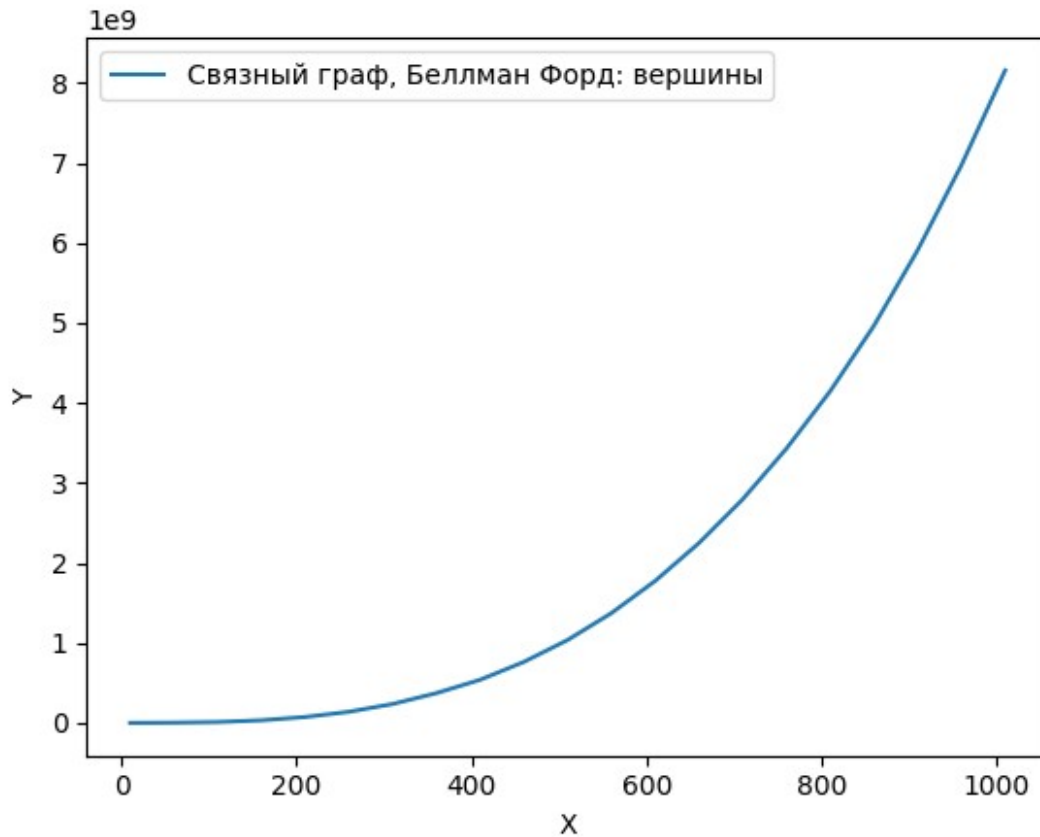
```
import pandas as pd
import matplotlib.pyplot as plt

# чтение csv файла
df = pd.read_csv('2_vertexes_bellman_ford.csv', sep=',')

# создание графика
plt.plot(df['x'], df['y'], label='Связный граф, Беллман Форд:
вершины')

# настройка осей и легенды
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()

# отображение графика
plt.show()
```



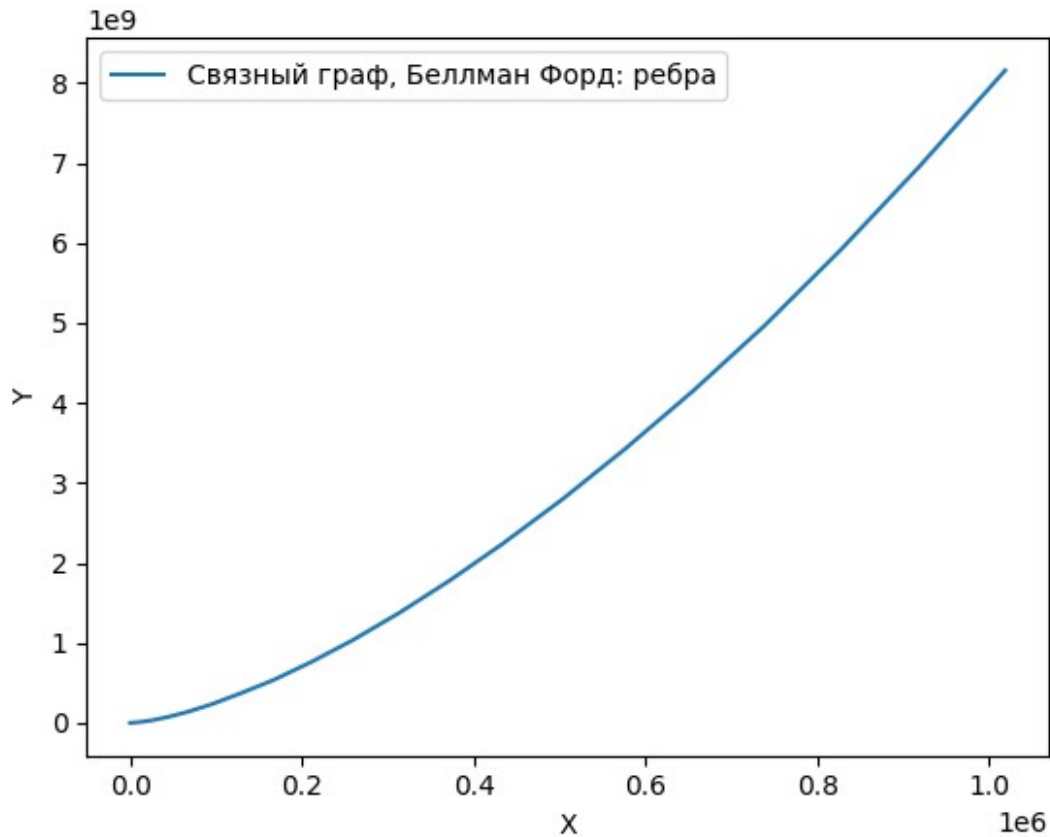
```
import pandas as pd
import matplotlib.pyplot as plt

# чтение csv файла
df = pd.read_csv('2_edges_bellman_ford.csv', sep=',')

# создание графика
plt.plot(df['x'], df['y'], label='Связный граф, Беллман Форд: ребра')

# настройка осей и легенды
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()

# отображение графика
plt.show()
```



РАЗРЕЖЕННЫЕ ГРАФЫ

Примечание: Для разреженного графа, в частности для дерева, количество ребер E зависит от числа вершин V по формуле:

$$E = V - 1$$

То бишь линейная зависимость, которая положительно влияет на асимптотику алгоритмов поиска (до этого зависимости были квадратичные, что ухудшало асимптотику).

С учетом этого в случае разреженных графов (вершины):

- Сложность Дейкстры:
 $O(n \log n)$
- Сложность Флойда-Уоршелла:
 $O(n^3)$
- Сложность Беллмана-Форда:
 $O(n^2)$

В случае разреженных графов (ребра):

- Сложность Дейкстры:
 $O(E \log E)$
- Сложность Флойда-Уоршелла:
 $O(E^3)$
- Сложность Беллмана-Форда:
 $O(E^2)$

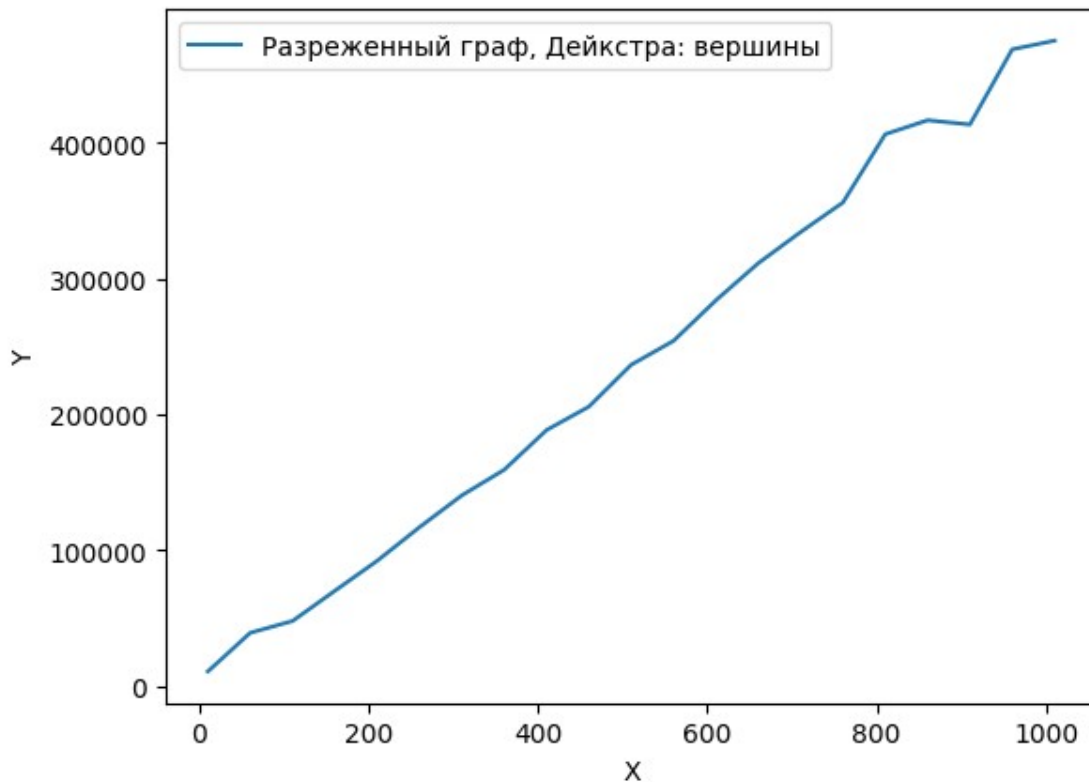
```
import pandas as pd
import matplotlib.pyplot as plt

# чтение csv файла
df = pd.read_csv('3_vertexes_dijkstra.csv', sep=',')

# создание графика
plt.plot(df['x'], df['y'], label='Разреженный граф, Дейкстра: вершины')

# настройка осей и легенды
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()

# отображение графика
plt.show()
```



```

import pandas as pd
import matplotlib.pyplot as plt

# чтение csv файла
df = pd.read_csv('3_edges_dijkstra.csv', sep=',')

# создание графика
plt.plot(df['x'], df['y'], label='Разреженный граф, Дейкстра: ребра')

# настройка осей и легенды
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()

# отображение графика
plt.show()

```



```

import pandas as pd
import matplotlib.pyplot as plt

# чтение csv файла
df = pd.read_csv('3_vertexes_floyd_warshall.csv', sep=',')

# создание графика
plt.plot(df['x'], df['y'], label='Разреженный граф, Флойд Уоршелл:')

```

```
вершины')
```

```
# настройка осей и легенды
```

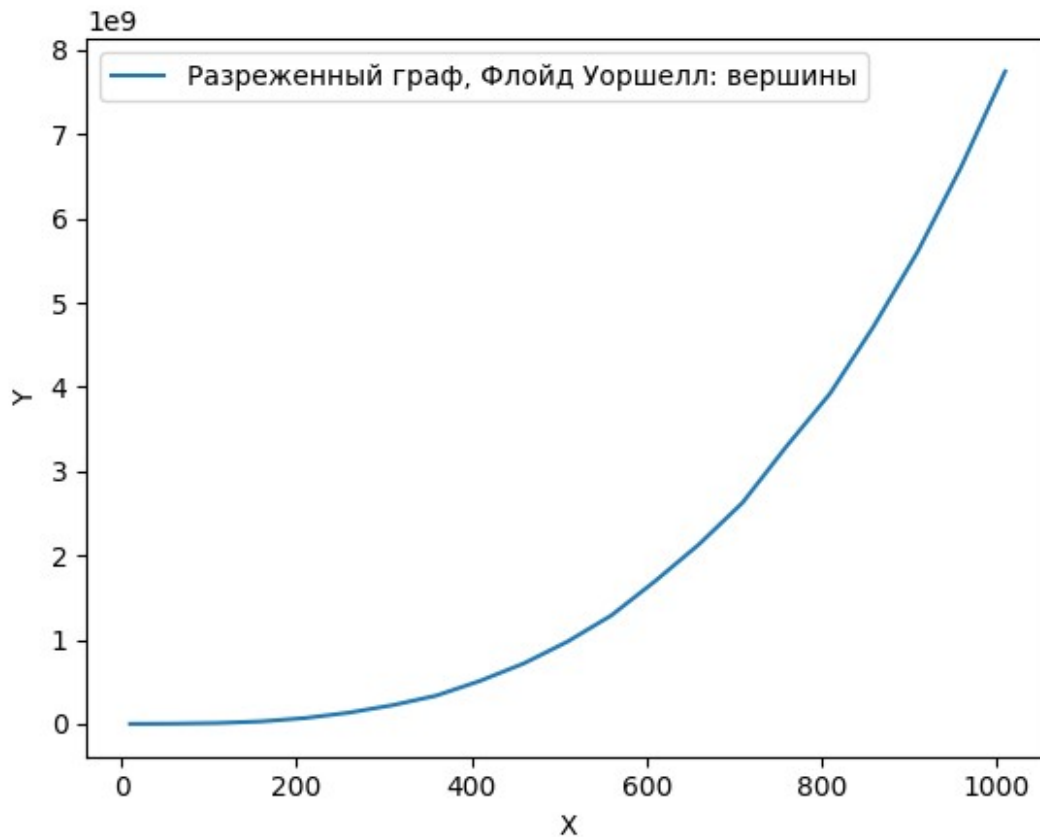
```
plt.xlabel('X')
```

```
plt.ylabel('Y')
```

```
plt.legend()
```

```
# отображение графика
```

```
plt.show()
```



```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
# чтение csv файла
```

```
df = pd.read_csv('3_edges_floyd_warshall.csv', sep=',')
```

```
# создание графика
```

```
plt.plot(df['x'], df['y'], label='Разреженный граф, Флойд Уоршелл:  
ребра')
```

```
# настройка осей и легенды
```

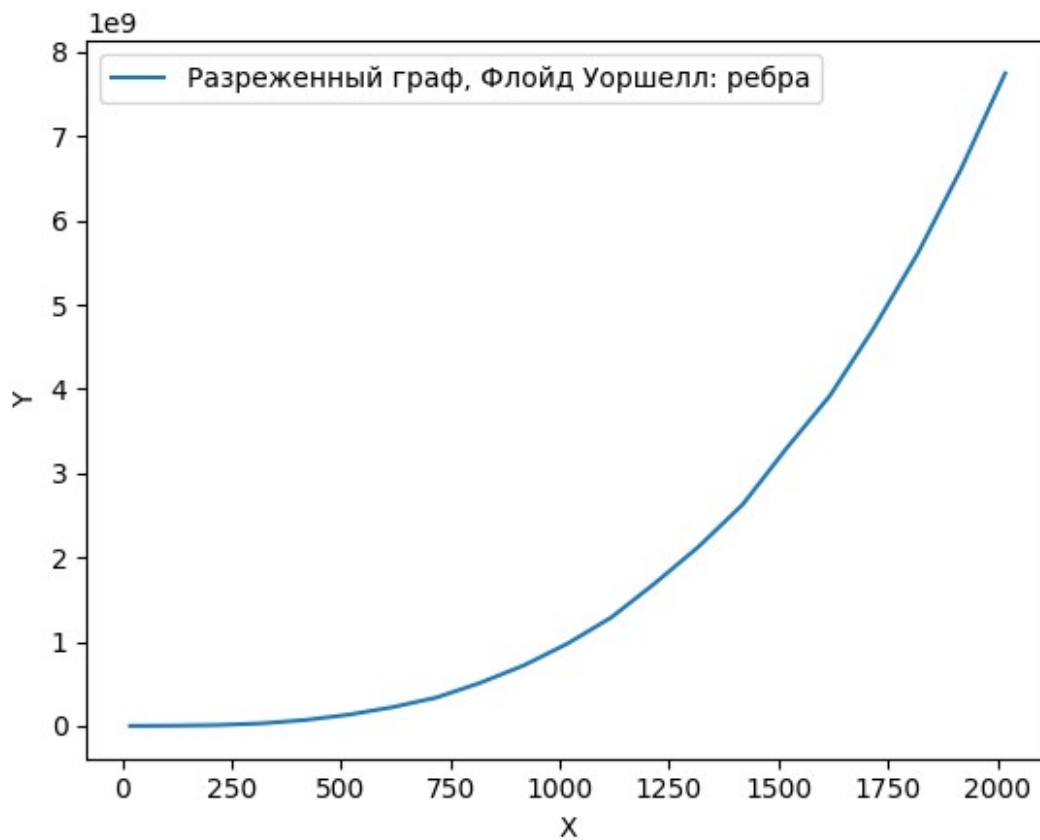
```
plt.xlabel('X')
```

```
plt.ylabel('Y')
```

```
plt.legend()
```



```
# отображение графика
plt.show()
```



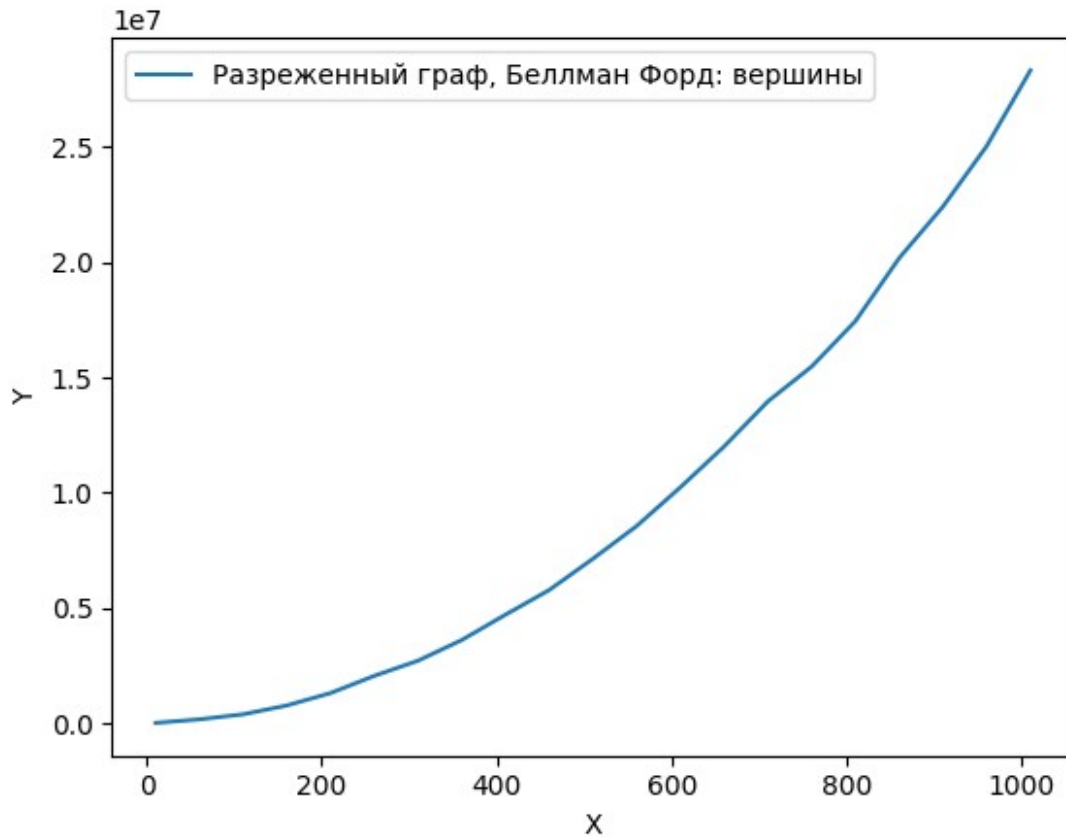
```
import pandas as pd
import matplotlib.pyplot as plt

# чтение csv файла
df = pd.read_csv('3_vertexes_bellman_ford.csv', sep=',')

# создание графика
plt.plot(df['x'], df['y'], label='Разреженный граф, Беллман Форд:
вершины')

# настройка осей и легенды
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()

# отображение графика
plt.show()
```



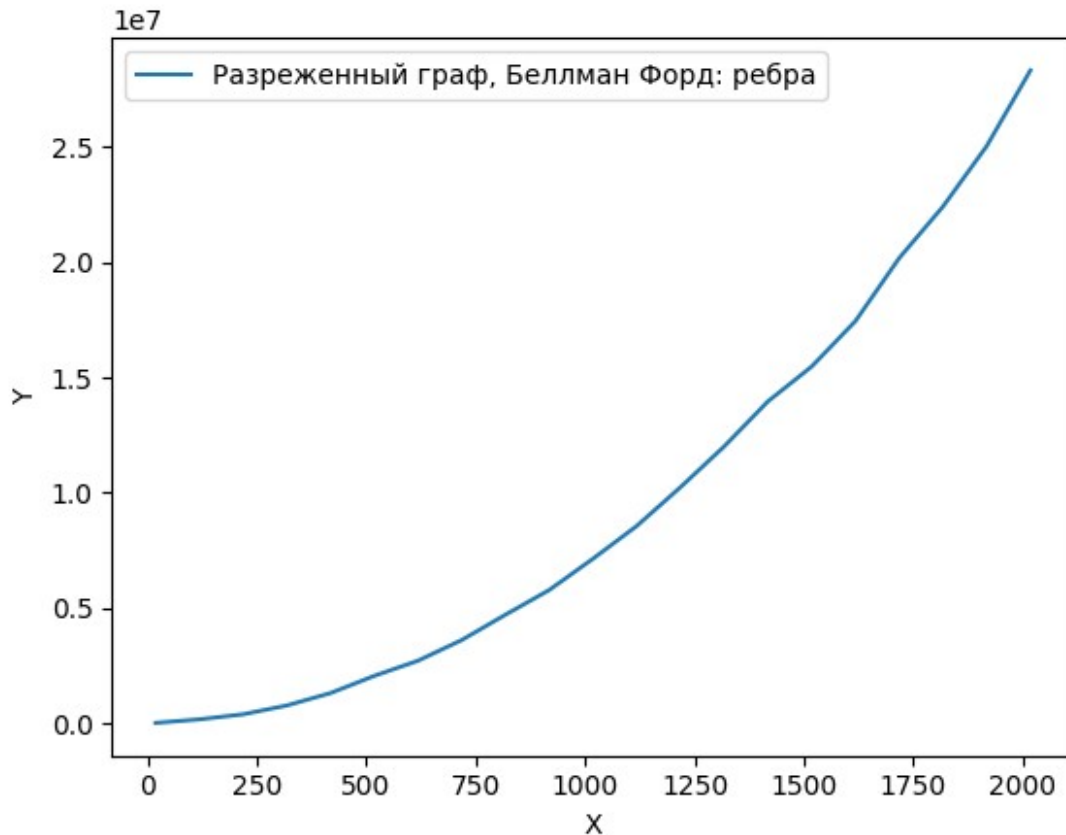
```
import pandas as pd
import matplotlib.pyplot as plt

# чтение csv файла
df = pd.read_csv('3_edges_bellman_ford.csv', sep=',')

# создание графика
plt.plot(df['x'], df['y'], label='Разреженный граф, Беллман Форд:
ребра')

# настройка осей и легенды
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()

# отображение графика
plt.show()
```



ОБЩИЕ ВЫВОДЫ по 1-ой части

На каждом графике при аппроксимации прослеживается соответствующая сложность алгоритма. Теория работает. И это прекрасно.

Агрегированные графики зависимости

```
import pandas as pd
import matplotlib.pyplot as plt

# чтение csv файла
df = pd.read_csv('full_graph_vertexes.csv', sep=',')

# создание графика
plt.plot(df['x'], df['y1'], color='red', label='Полный граф, Дейкстра: вершины')
plt.plot(df['x'], df['y2'], color='blue', label='Полный граф, Флойд Уоршелл: вершины')
plt.plot(df['x'], df['y3'], color='green', label='Полный граф, Беллман
```

```
Форд: вершины')
```

```
# настройка осей и легенды
```

```
plt.xlabel('X')
```

```
plt.ylabel('Y')
```

```
plt.legend()
```

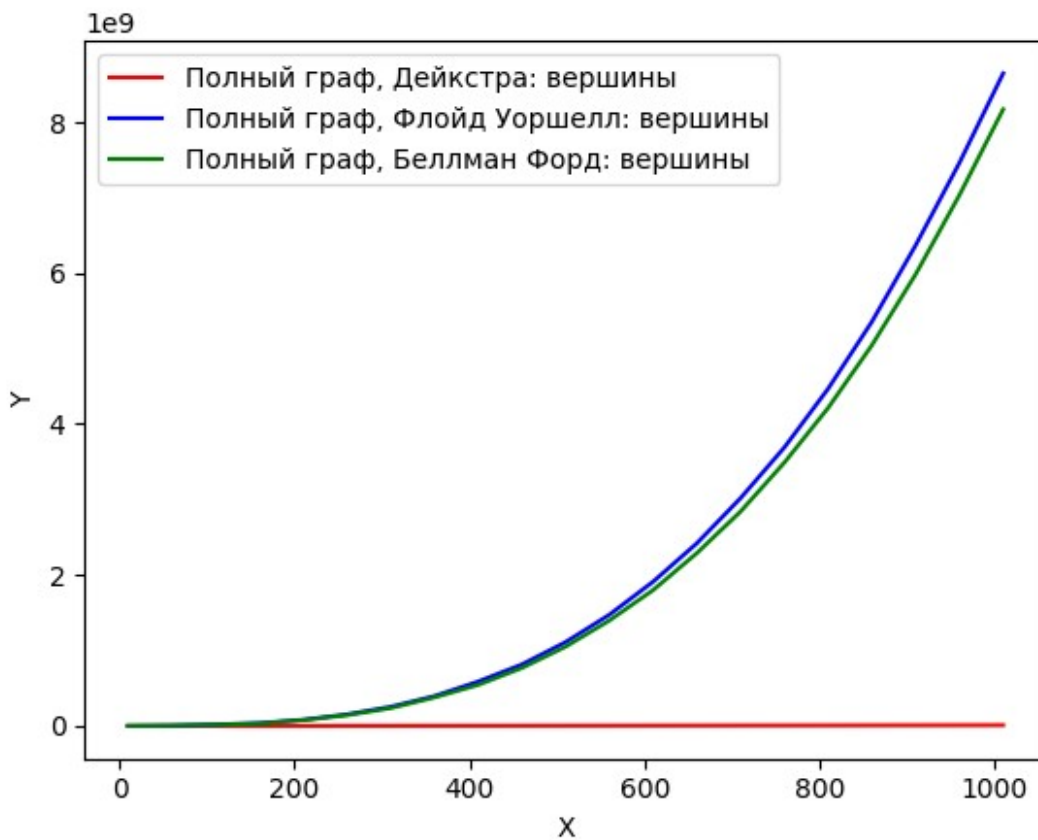
```
# отображение графика
```

```
plt.show()
```

```
### КОММЕНТАРИЙ:
```

```
# На больших данных явно видно, что сложность  $n^2 * \log n$  значительно  
лучше
```

```
# кубической плотности. Дейкстра рулит!
```



```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
# чтение csv файла
```

```
df = pd.read_csv('full_graph_edges.csv', sep=',')
```

```
# создание графика
```

```
plt.plot(df['x'], df['y1'], color='red', label='Полный граф, Дейкстра:
```

```

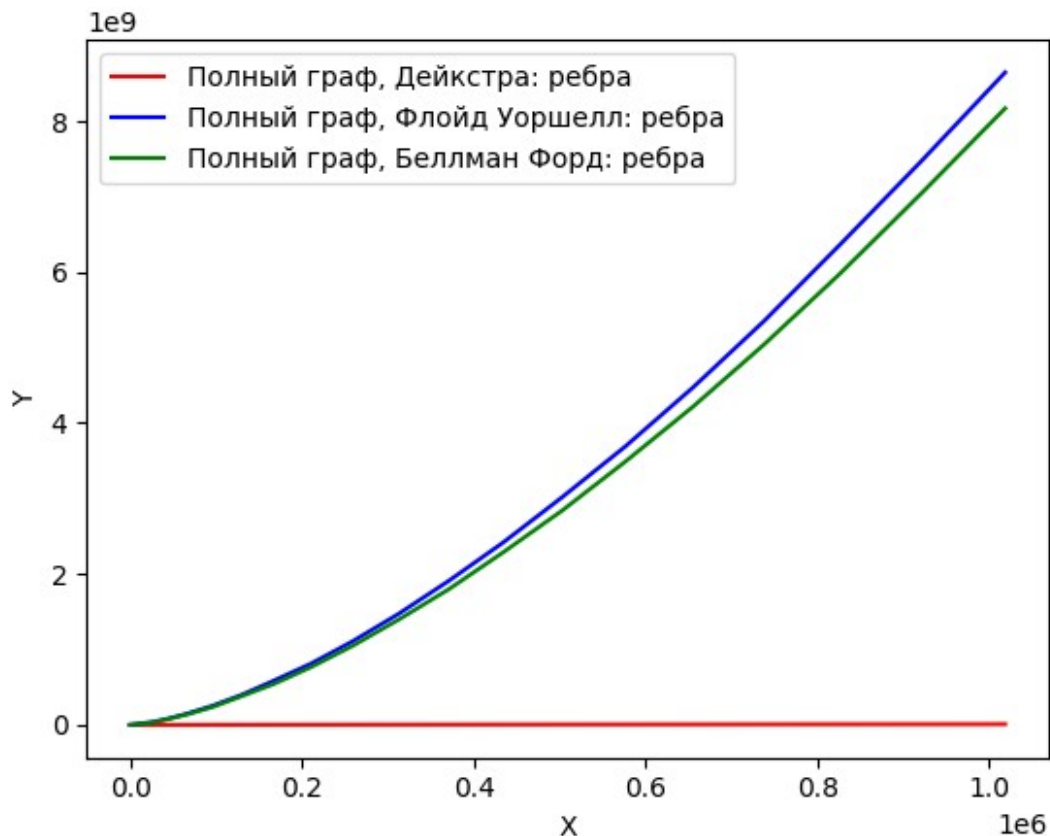
ребра')
plt.plot(df['x'], df['y2'], color='blue', label='Полный граф, Флойд
Уоршелл: ребра')
plt.plot(df['x'], df['y3'], color='green', label='Полный граф, Беллман
Форд: ребра')

# настройка осей и легенды
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()

# отображение графика
plt.show()

```

КОММЕНТАРИЙ:
 # На больших данных явно видно, что сложность $E * \log E$ значительно
 лучше
 # плотности $E \sim 1.5$. Дейкстра рулит!



```

import pandas as pd
import matplotlib.pyplot as plt

```

чтение csv файла

```

df = pd.read_csv('connected_graph_vertexes.csv', sep=',')

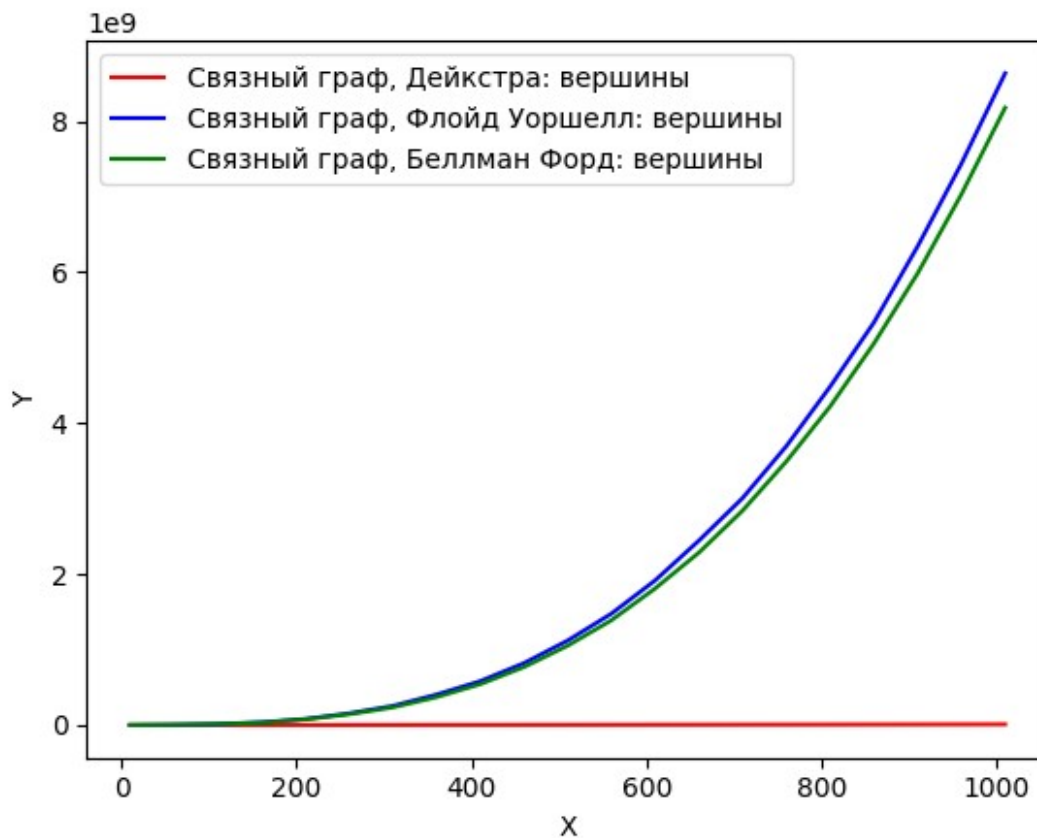
# создание графика
plt.plot(df['x'], df['y1'], color='red', label='Связный граф, Дейкстра: вершины')
plt.plot(df['x'], df['y2'], color='blue', label='Связный граф, Флойд Уоршелл: вершины')
plt.plot(df['x'], df['y3'], color='green', label='Связный граф, Беллман Форд: вершины')

# настройка осей и легенды
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()

# отображение графика
plt.show()

### КОММЕНТАРИЙ
# Аналогично случаю с полным графом (см выше)

```



```

import pandas as pd
import matplotlib.pyplot as plt

```

```

# чтение csv файла
df = pd.read_csv('connected_graph_edges.csv', sep=',')

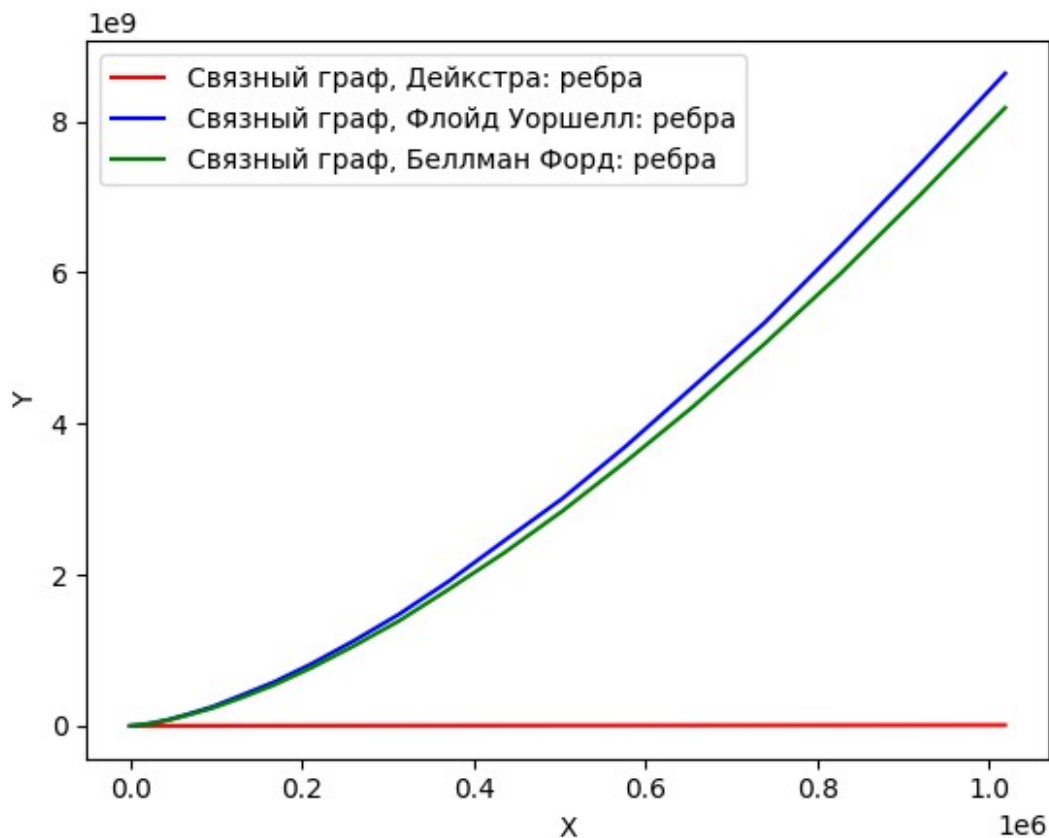
# создание графика
plt.plot(df['x'], df['y1'], color='red', label='Связный граф, Дейкстра: ребра')
plt.plot(df['x'], df['y2'], color='blue', label='Связный граф, Флойд Уоршелл: ребра')
plt.plot(df['x'], df['y3'], color='green', label='Связный граф, Беллман Форд: ребра')

# настройка осей и легенды
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()

# отображение графика
plt.show()

### КОММЕНТАРИЙ
# Аналогично случаю с полным графом (см выше)

```



```
import pandas as pd
import matplotlib.pyplot as plt

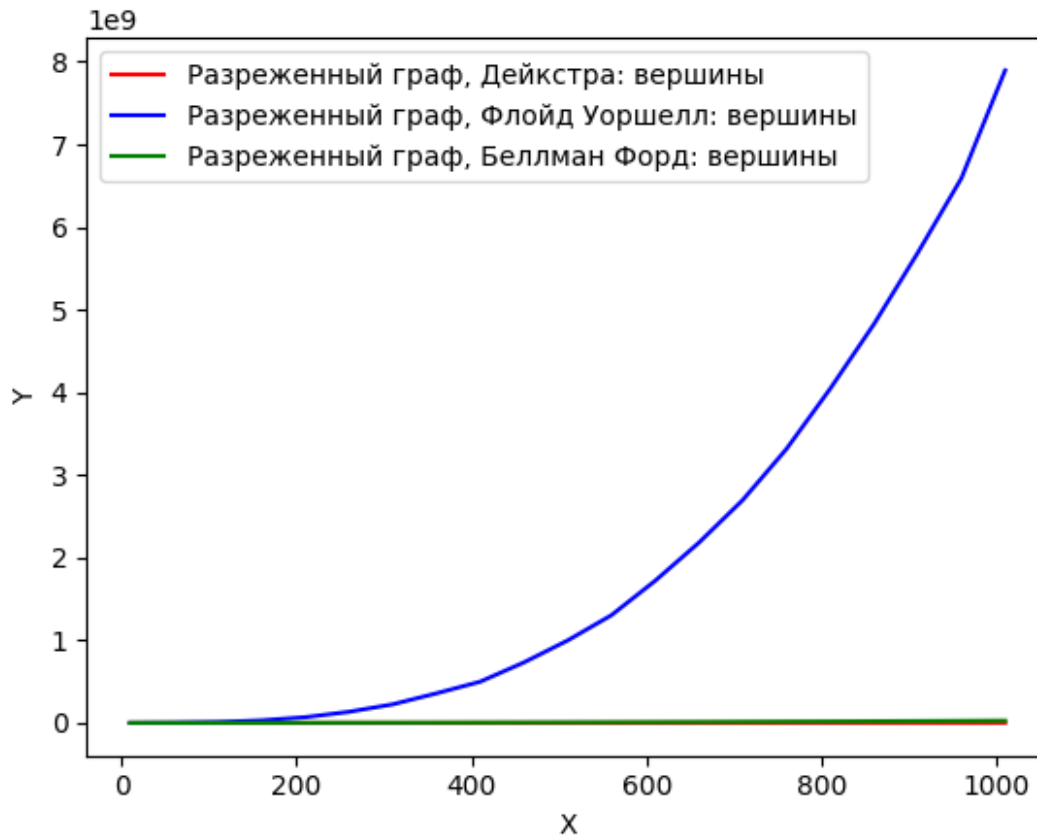
# чтение csv файла
df = pd.read_csv('sparse_graph_vertexes.csv', sep=',')

# создание графика
plt.plot(df['x'], df['y1'], color='red', label='Разреженный граф,
Дейкстра: вершины')
plt.plot(df['x'], df['y2'], color='blue', label='Разреженный граф,
Фloyd Уоршелл: вершины')
plt.plot(df['x'], df['y3'], color='green', label='Разреженный граф,
Беллман Форд: вершины')

# настройка осей и легенды
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()

# отображение графика
plt.show()

### КОММЕНТАРИЙ
# Дейкстра( $E * \log E$ ) и Беллман-Форд( $E^2$ ) работают намного лучше
# Флойда-Уоршелла( $E^3$ )!
```

```
import pandas as pd
import matplotlib.pyplot as plt

# чтение csv файла
df = pd.read_csv('sparse_graph_edges.csv', sep=',')

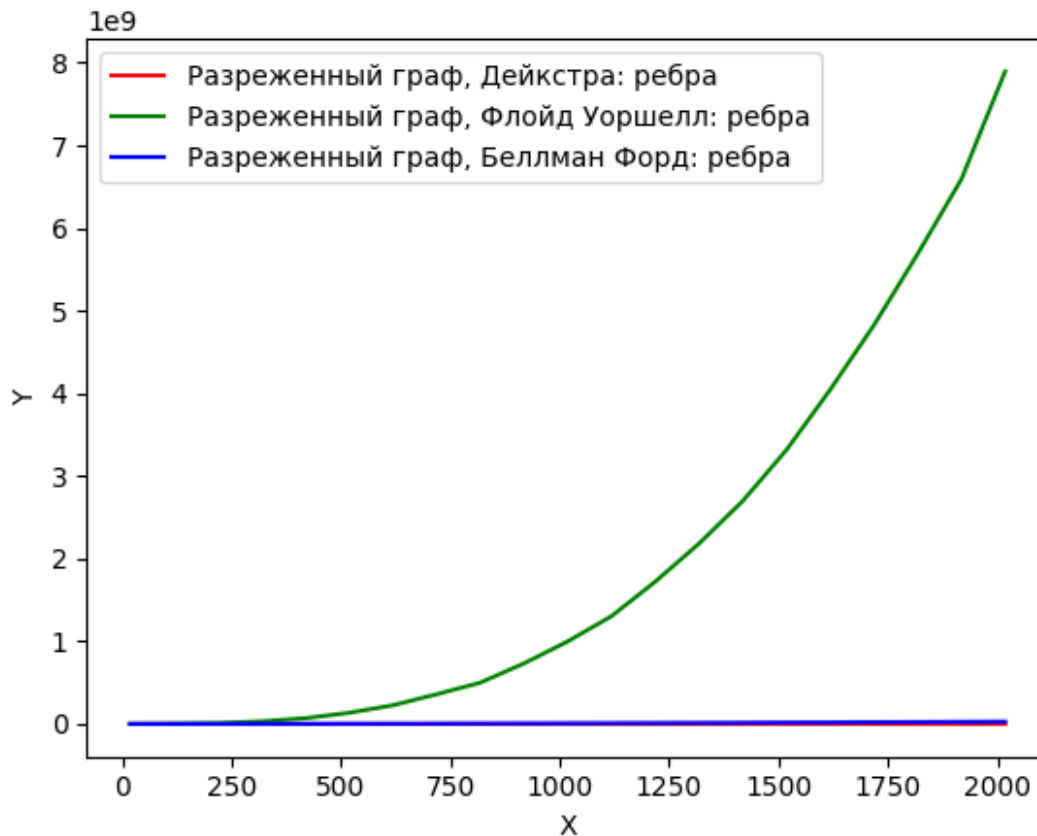
# создание графика
plt.plot(df['x'], df['y1'], color='red', label='Разреженный граф, Дейкстра: ребра')
plt.plot(df['x'], df['y2'], color='green', label='Разреженный граф, Флойд Уоршелл: ребра')
plt.plot(df['x'], df['y3'], color='blue', label='Разреженный граф, Беллман Форд: ребра')

# настройка осей и легенды
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()

# отображение графика
plt.show()
```

КОММЕНТАРИЙ

Дейкстра($E * \log E$) и Беллман-Форд(E^2) работают намного лучше Флойда-Уоршелла(E^3)!



ВЫВОДЫ ПО ВТОРОЙ ЧАСТИ:

На полном и связном графах лучше всего работает алгоритм Дейкстры. На разреженном графе (дереве) лучше всего работают алгоритмы Дейкстры и Беллмана Форда. Обоснование тому - подтвержденная на практике теория, изложенная в самом начале данного блокнота.

Мария Константиновна, большое Вам спасибо за курс по алгоритмам! За этот учебный год я преисполнился в познаниях, научился решать сложные задачи и обосновывать свои решения, а также развил алгоритмическое мышление! До новых встреч!

НЕ МОГУ НЕ ПОЗДРАВИТЬ Манула Тимофея с 3-летием!

Пару дней назад любимцу Московского зоопрака Тимоше исполнилось 3 года. За эти 3 года он здорово вырос, окреп и набрался сил. Пожелаем ему здоровья, наслаждения сочными королевскими крысками и возможности

продолжить свой род! Ура!

