

Discrete.py d1.py Dis.py Plots

7 class GraphVisualizer: 1 10 3 8 47

def measure\_and\_plot\_time(self, si

81 # Побудова графіка часу

82 plt.plot(\*args: sizes, matrix\_t

83 plt.plot(\*args: sizes, list\_tim

84 plt.xlabel("Кількість вершин (

85 plt.ylabel("Час виконання (сек

86 plt.title(f"Аналіз часу побудо

87 plt.legend()

88 plt.grid(True)

89 plt.show()

90

91 num\_nodes = 10

92 probability = 0.7

93

94 # Створюємо граф для демонстрації

95 builder = GraphBuilder(num\_nodes, prob

96 print("Edges:", builder.get\_edges())

97

98

640x480 PNG (32-bit color) 59,5 kB

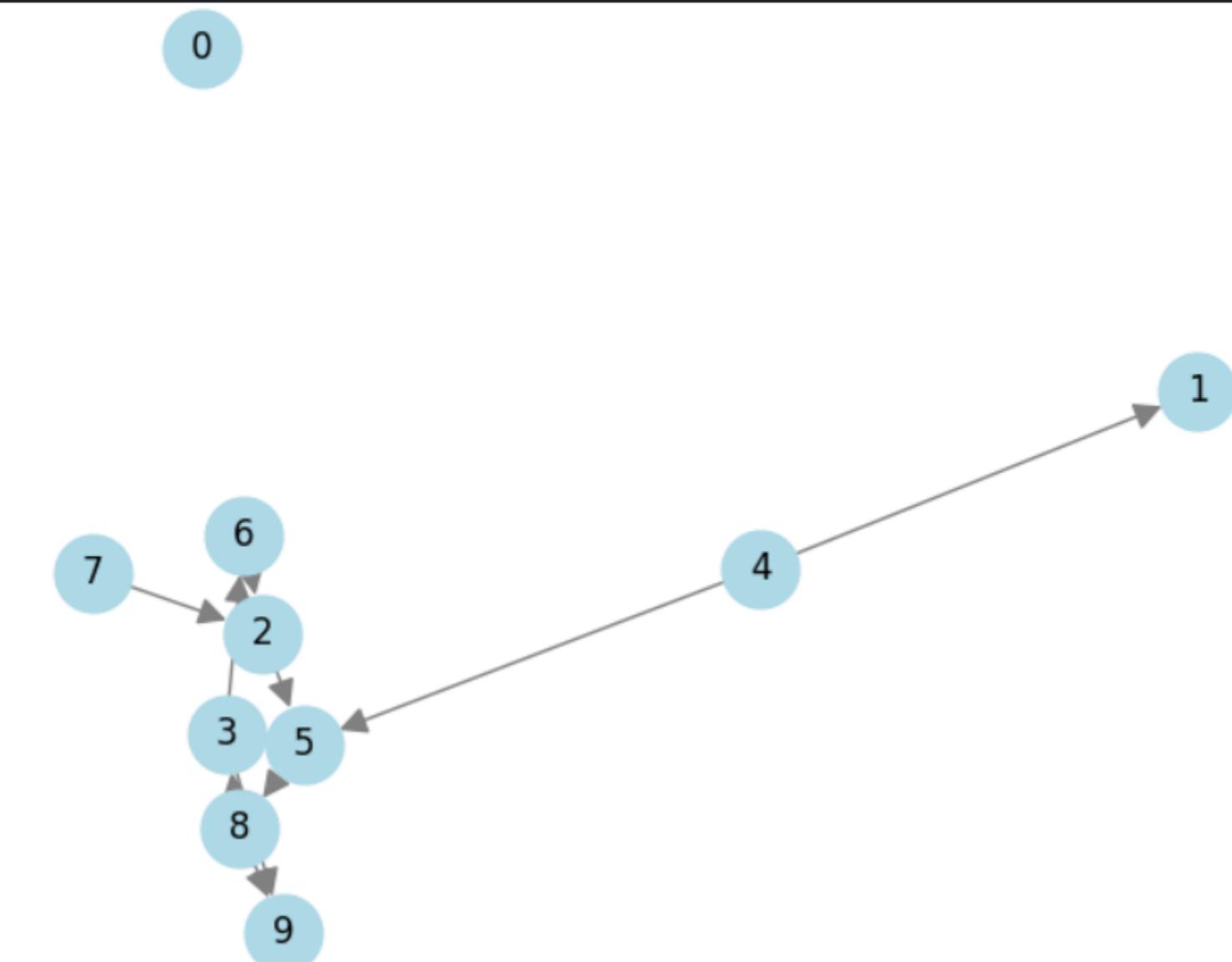
```
graph TD; 3 --> 0; 3 --> 1; 3 --> 2; 3 --> 4; 3 --> 5; 3 --> 6; 3 --> 7; 3 --> 8; 3 --> 9; 6 --> 0; 6 --> 1; 6 --> 2; 6 --> 4; 6 --> 5; 6 --> 7; 6 --> 8; 6 --> 9; 0 --> 1; 0 --> 4; 0 --> 5; 0 --> 7; 0 --> 8; 0 --> 9; 1 --> 2; 1 --> 4; 1 --> 5; 1 --> 7; 1 --> 8; 1 --> 9; 2 --> 4; 2 --> 5; 2 --> 7; 2 --> 8; 2 --> 9; 4 --> 5; 4 --> 7; 4 --> 8; 4 --> 9; 5 --> 7; 5 --> 8; 5 --> 9; 7 --> 8; 7 --> 9; 8 --> 9;
```

```
7 class GraphVisualizer: 10 3 1: 10 3 1: 10 3 1:
```

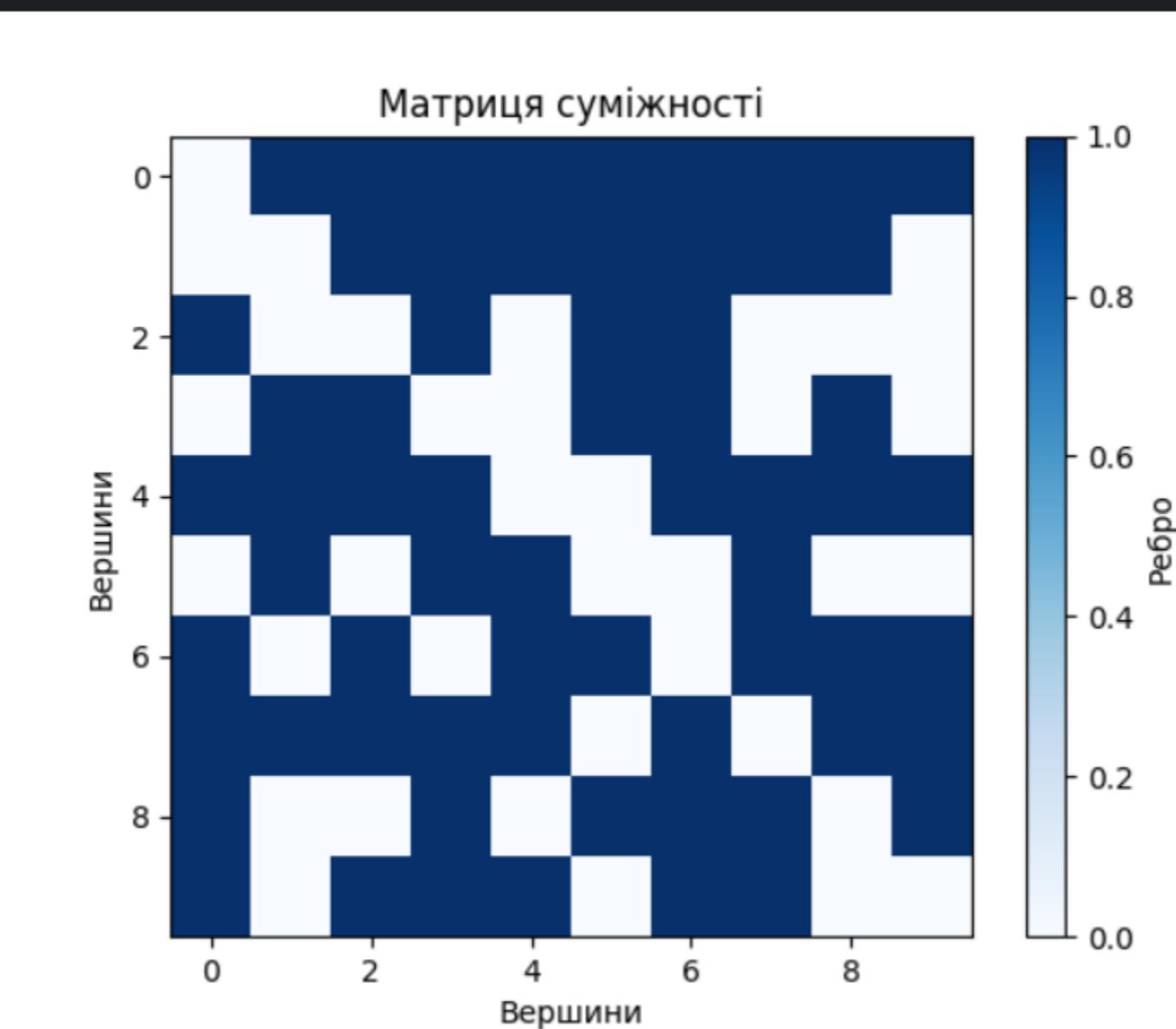
```
47     def measure_and_plot_time(self, si:
```

```
81         # Побудова графіка часу
82         plt.plot(*args: sizes, matrix_t)
83         plt.plot(*args: sizes, list_tim
84         plt.xlabel("Кількість вершин (n")
85         plt.ylabel("Час виконання (секунди")
86         plt.title(f"Аналіз часу побудо
87         plt.legend()
88         plt.grid(True)
89         plt.show()
```

```
90
91 num_nodes = 10
92 probability = 0.7
93
94
95 # Створюємо граф для демонстрації
96 builder = GraphBuilder(num_nodes, probability)
97 print("Edges:", builder.get_edges())
98
```

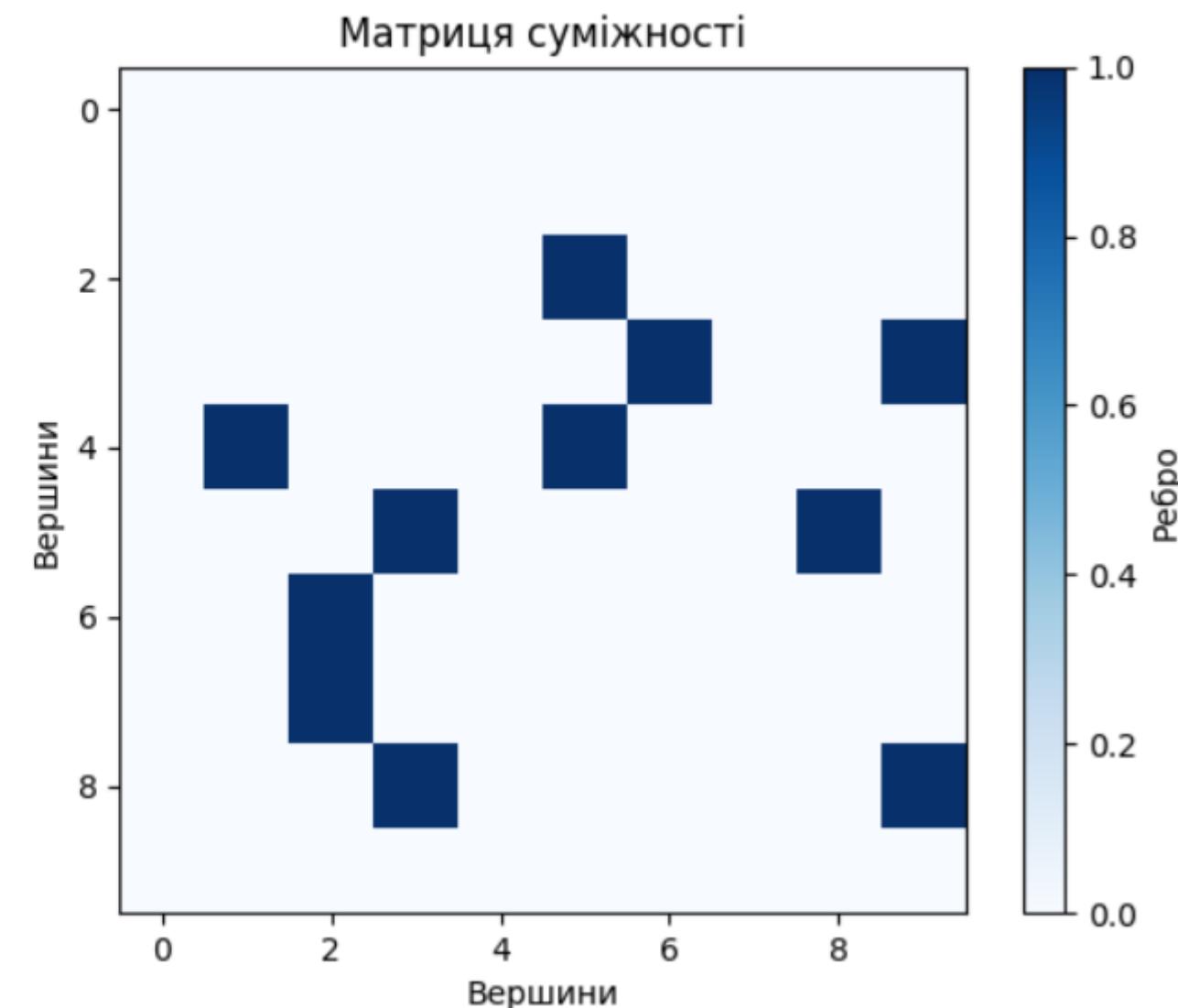


```
7 class GraphVisualizer: 1 🔍 10 ✅ 3 ⌂
47     def measure_and_plot_time(self, si
81         # Побудова графіка часу
82         plt.plot(*args: sizes, matrix_t
83         plt.plot(*args: sizes, list_tim
84         plt.xlabel("Кількість вершин (n)
85         plt.ylabel("Час виконання (секунди)
86         plt.title(f"Аналіз часу побудови
87         plt.legend()
88         plt.grid(True)
89         plt.show()
90
91
92     num_nodes = 10
93     probability = 0.7
94
95     # Створюємо граф для демонстрації
96     builder = GraphBuilder(num_nodes, prob
97     print("Edges:", builder.get_edges())
98
```



Discrete.py d1.py Dis.py Plots

```
7     class GraphVisualizer: 10 10 3 3
47         def measure_and_plot_time(self, si
81             # Побудова графіка часу
82             plt.plot(*args: sizes, matrix_t
83             plt.plot(*args: sizes, list_tim
84             plt.xlabel("Кількість вершин (")
85             plt.ylabel("Час виконання (сек")
86             plt.title(f"Аналіз часу побудо
87             plt.legend()
88             plt.grid(True)
89             plt.show()
90
91
92         num_nodes = 10
93         probability = 0.1
94
95         # Створюємо граф для демонстрації
96         builder = GraphBuilder(num_nodes, prob
97         print("Edges:", builder.get_edges())
98
```



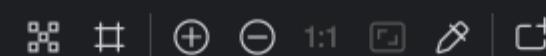
Discrete.py

d1.py

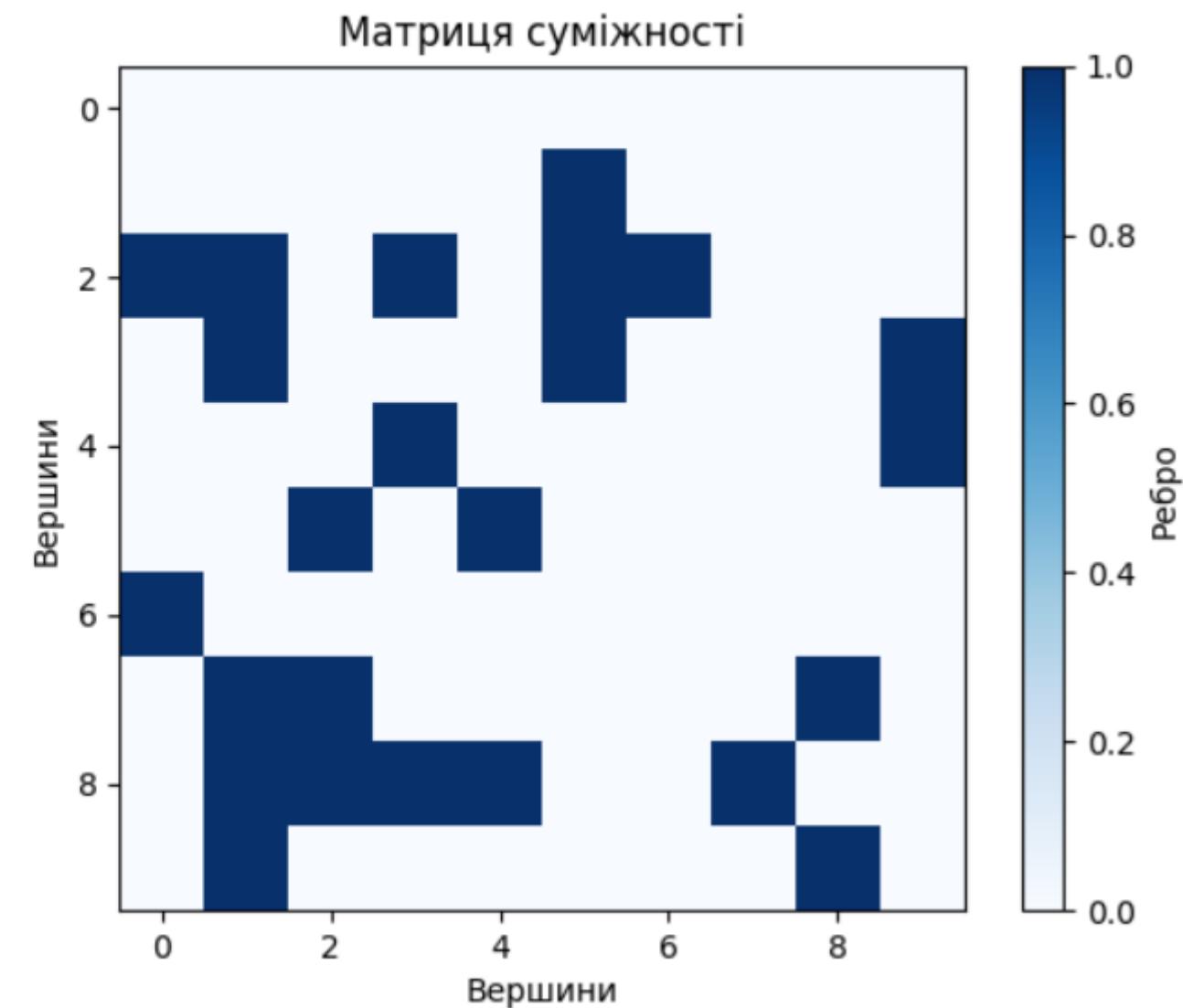
Dis.py

Plots

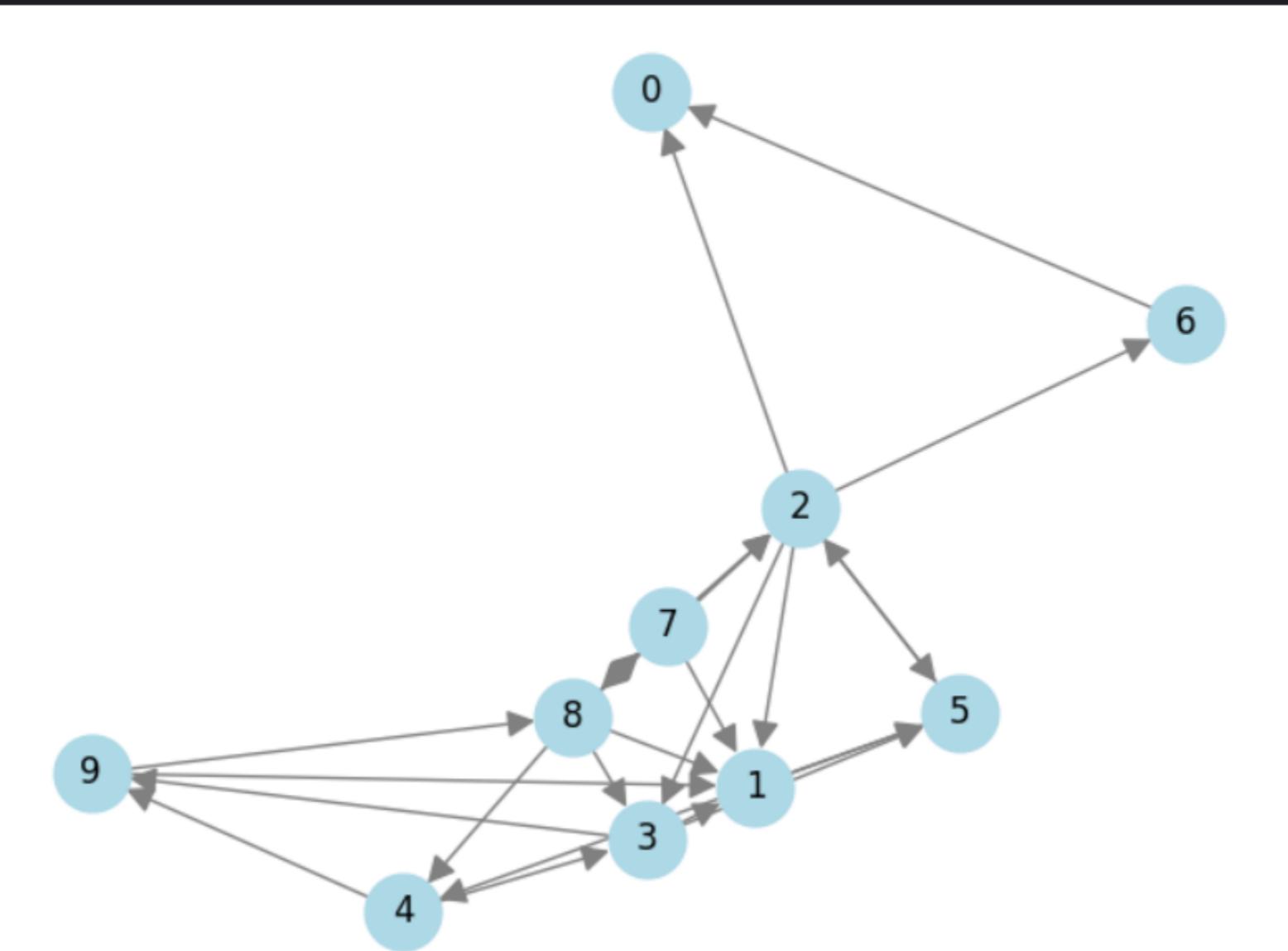
```
7     class GraphVisualizer: 1 10 3 ^ ~
47         def measure_and_plot_time(self, si
81             # Побудова графіка часу
82             plt.plot(*args: sizes, matrix_t
83             plt.plot(*args: sizes, list_tim
84             plt.xlabel("Кількість вершин (n)
85             plt.ylabel("Час виконання (сек)
86             plt.title(f"Аналіз часу побудо
87             plt.legend()
88             plt.grid(True)
89             plt.show()
90
91
92             num_nodes = 10
93             probability = 0.3
94
95             # Створюємо граф для демонстрації
96             builder = GraphBuilder(num_nodes, prob
97             print("Edges:", builder.get_edges())
98
```



640x480 PNG (32-bit color) 21,73 k



```
7     class GraphVisualizer: 1: 10 3  ^  v
 47         def measure_and_plot_time(self, si
81             # Побудова графіка часу
82             plt.plot(*args: sizes, matrix_t
83             plt.plot(*args: sizes, list_tim
84             plt.xlabel("Кількість вершин (n)
85             plt.ylabel("Час виконання (сек)
86             plt.title(f"Аналіз часу побудо
87             plt.legend()
88             plt.grid(True)
89             plt.show()
90
91
92     num_nodes = 10
93     probability = 0.3
94
95     # Створюємо граф для демонстрації
96     builder = GraphBuilder(num_nodes, prob
97     print("Edges:", builder.get_edges())
98
```



640x480 PNG (32-bit color) 28,05 kB

```
7 class GraphVisualizer: 1 10 3 ^ v
47     def measure_and_plot_time(self, si
81             # Побудова графіка часу
82             plt.plot(*args: sizes, matrix_t
83             plt.plot(*args: sizes, list_tim
84             plt.xlabel("Кількість вершин (n)
85             plt.ylabel("Час виконання (секунди)
86             plt.title(f"Аналіз часу побудови
87             plt.legend()
88             plt.grid(True)
89             plt.show()
90
91
92         num_nodes = 10
93         probability = 0.1
94
95         # Створюємо граф для демонстрації
96         builder = GraphBuilder(num_nodes, prob
97         print("Edges:", builder.get_edges())
```

