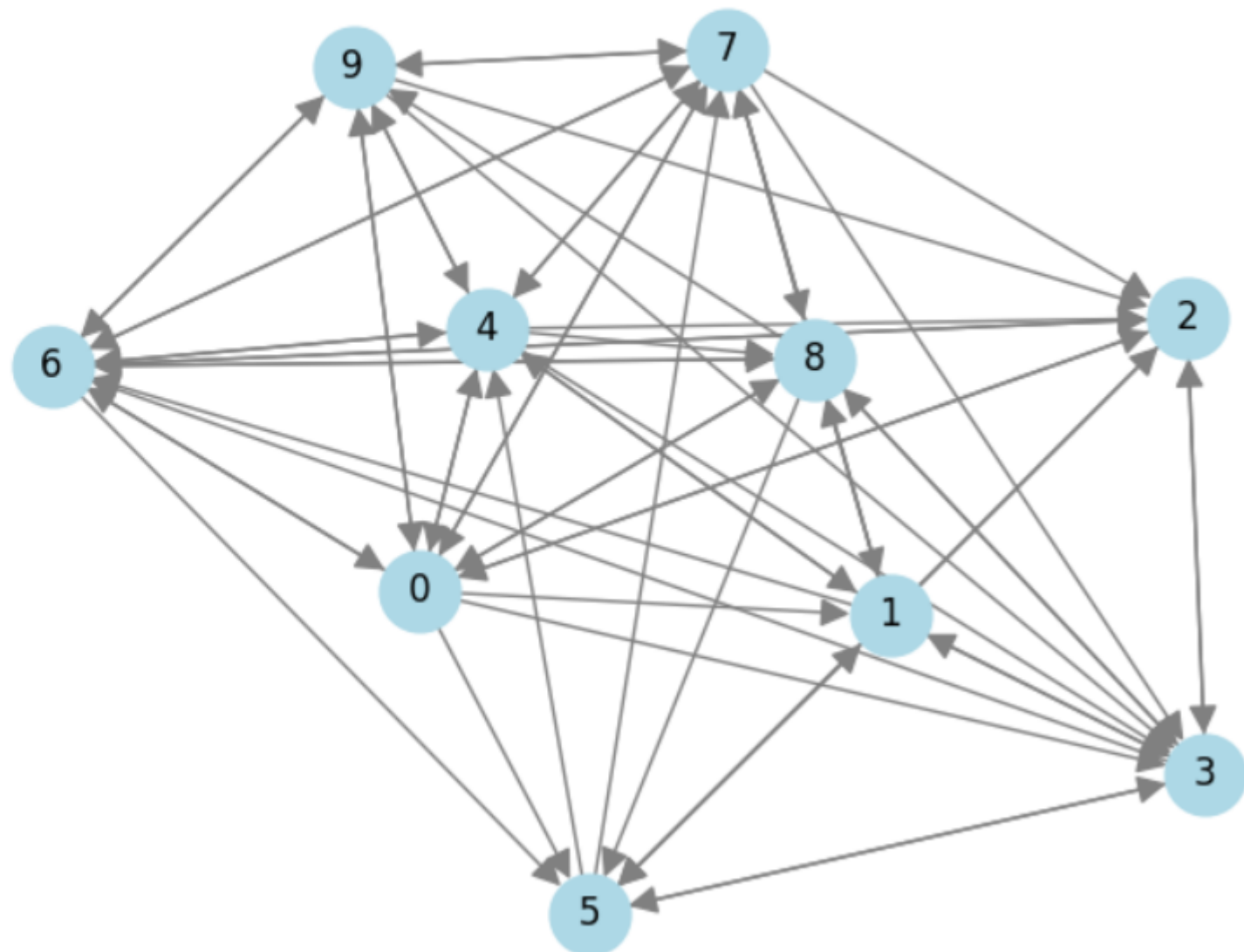
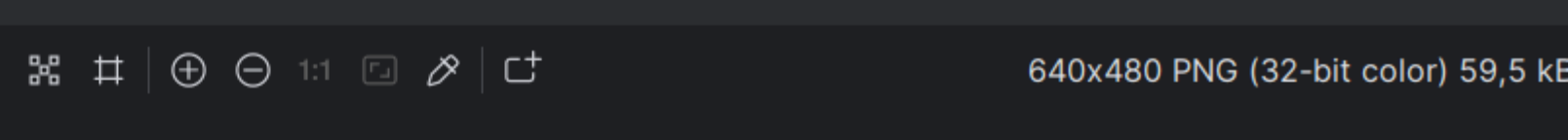


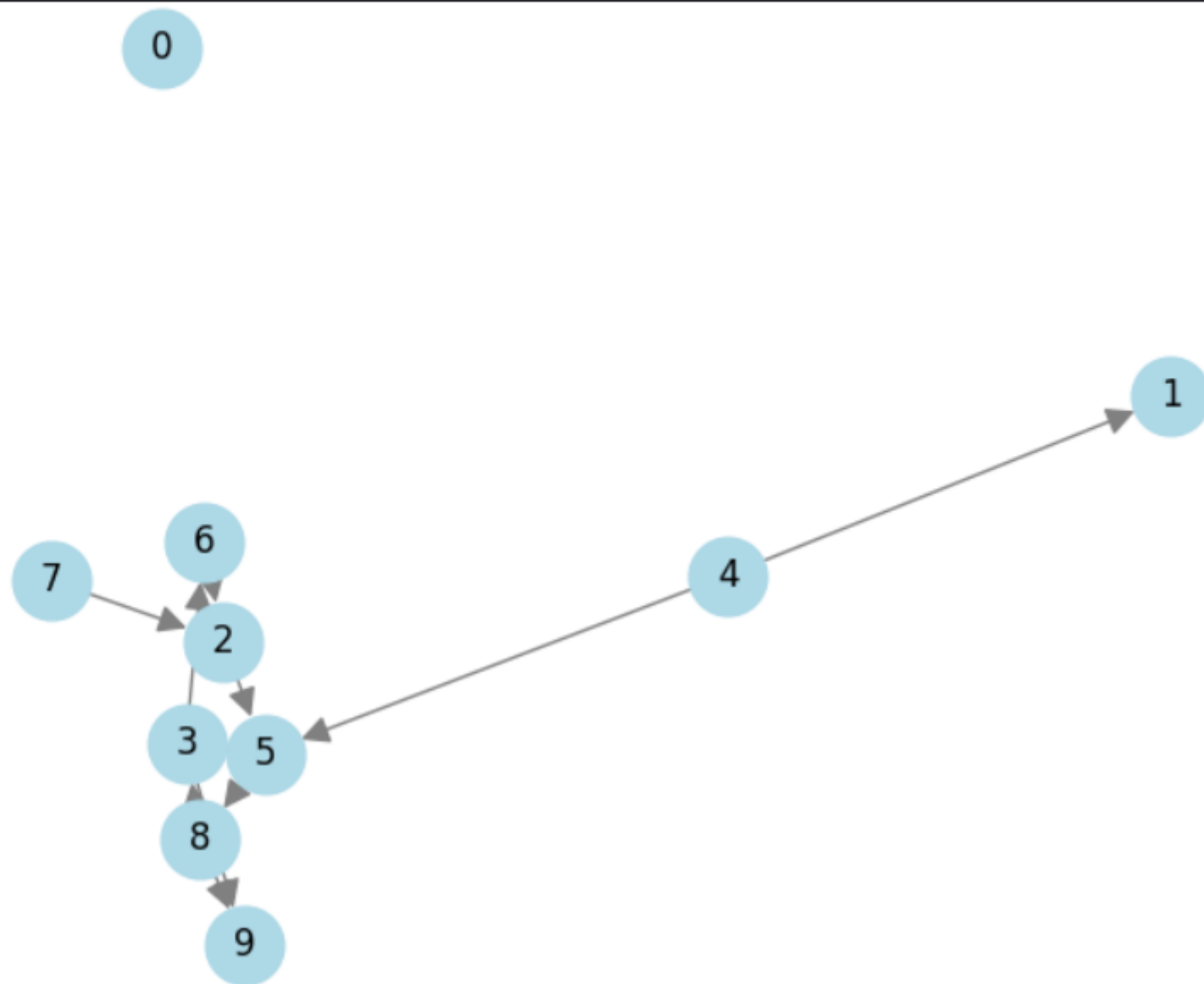
```

7      class GraphVisualizer:
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 = 0.7
94
95          # Створюємо граф для демонстрації
96          builder = GraphBuilder(num_nodes, prob
97          print("Edges:", builder.get_edges())
98

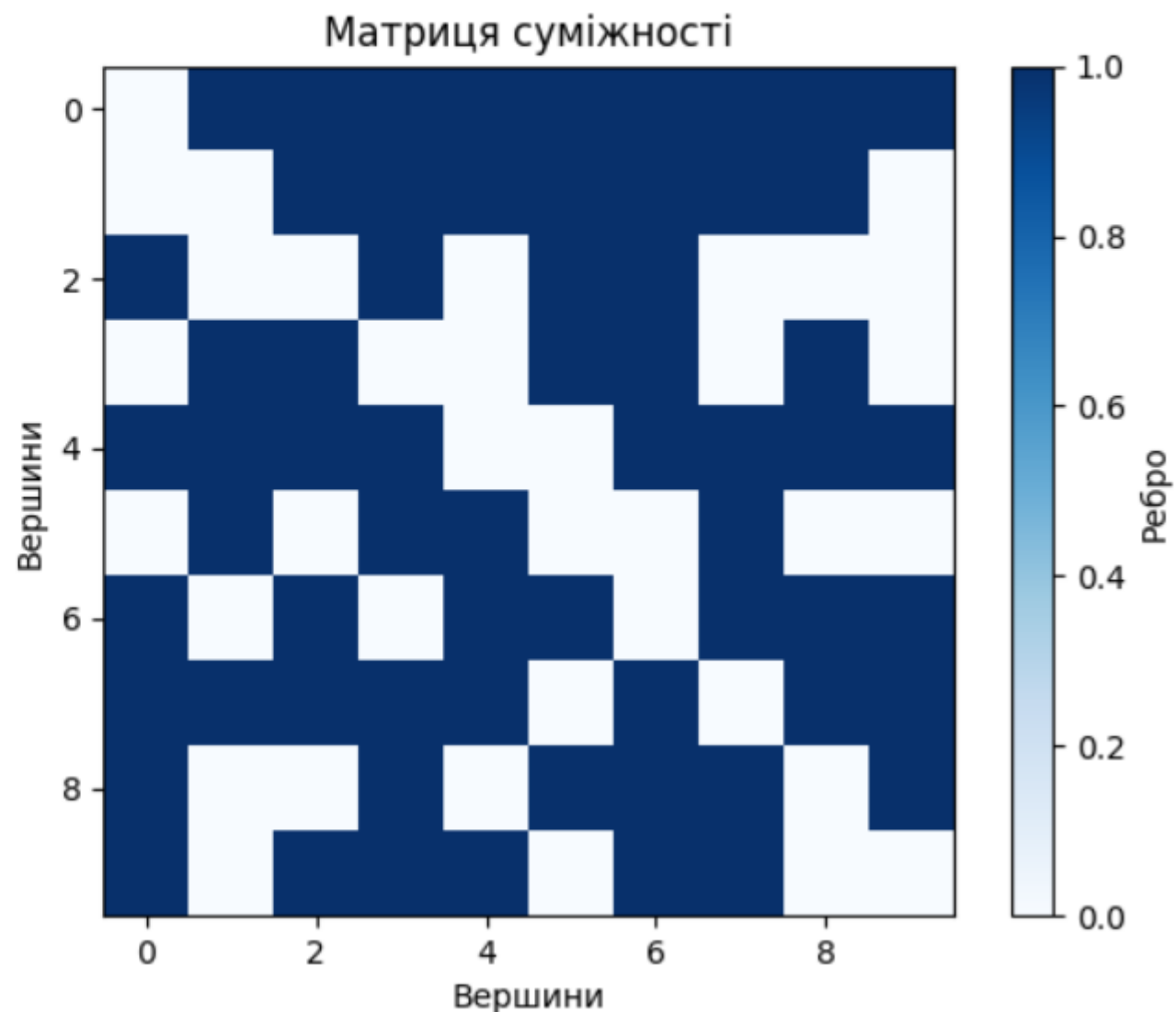
```



```
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 (
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 = 0.7
94
95 # Створюємо граф для демонстрації
96 builder = GraphBuilder(num_nodes, prob
97 print("Edges:", builder.get_edges())
98
```



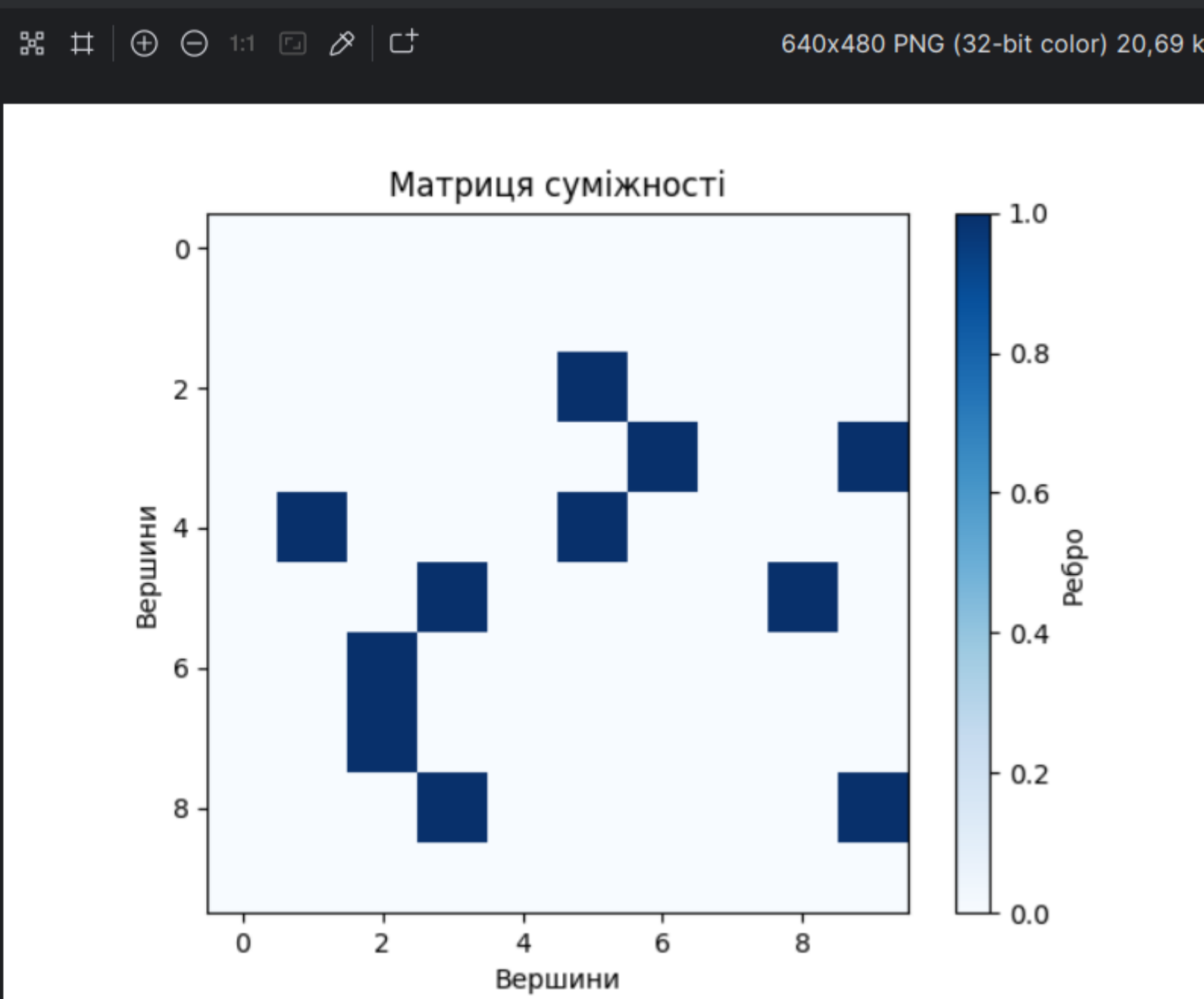
```
7 class GraphVisualizer: 1 | 10 3 ^ v
47 def measure_and_plot_time(self, si
81 # Побудова графіка часу
82 plt.plot(*args: sizes, matrix_t
83 plt.plot(*args: sizes, list_tim
84 plt.xlabel("Кількість вершин (
85 plt.ylabel("Час виконання (сек
86 plt.title(f"Аналіз часу побудо
87 plt.legend()
88 plt.grid(True)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



```

7      class GraphVisualizer:
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

```



```
7 class GraphVisualizer:
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()
```

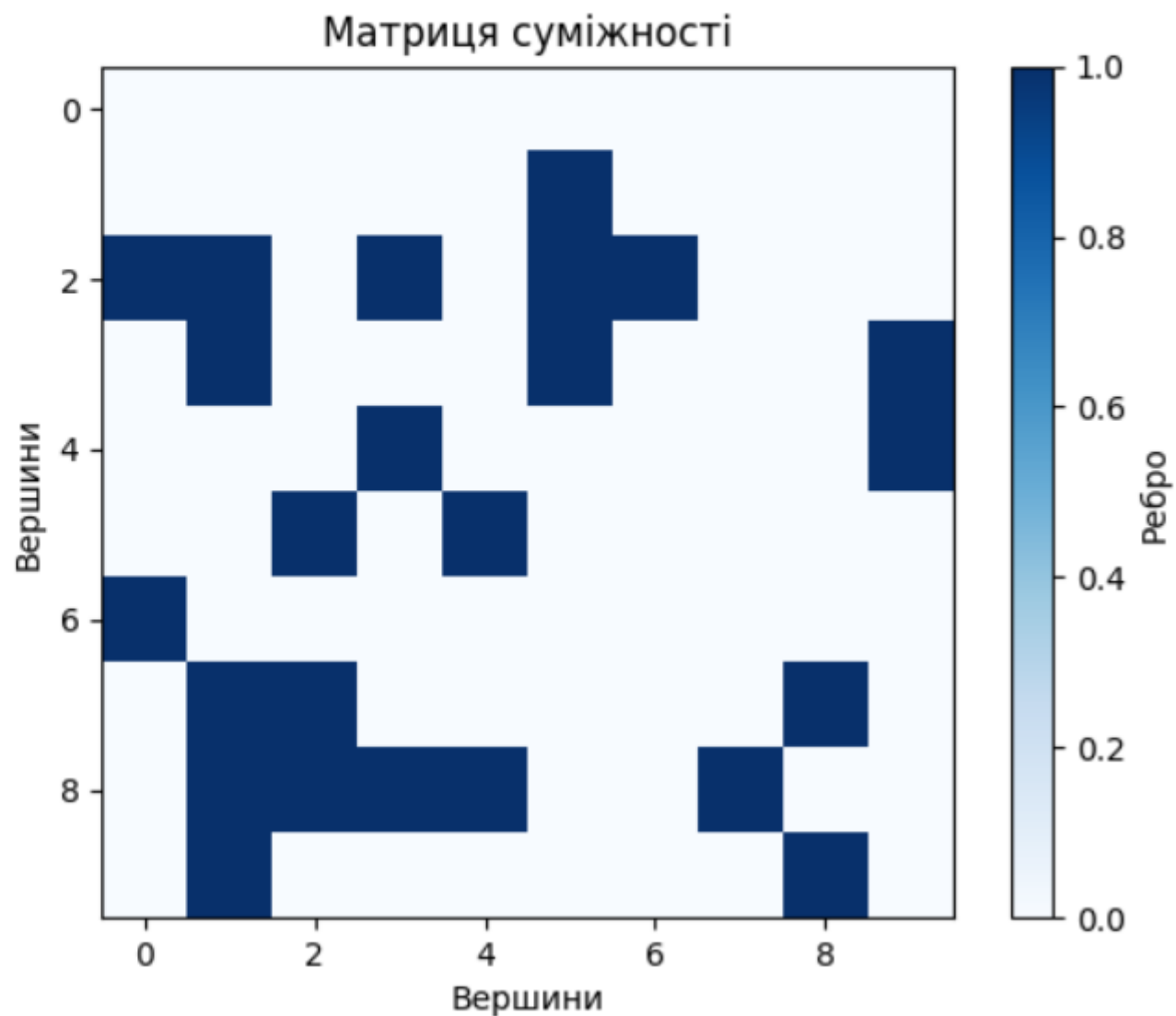
```
92     num_nodes = 10
```

```
93     probability = 0.3
```

```
95     # Створюємо граф для демонстрації
```

```
96     builder = GraphBuilder(num_nodes, prob
```

```
97     print("Edges:", builder.get_edges())
```

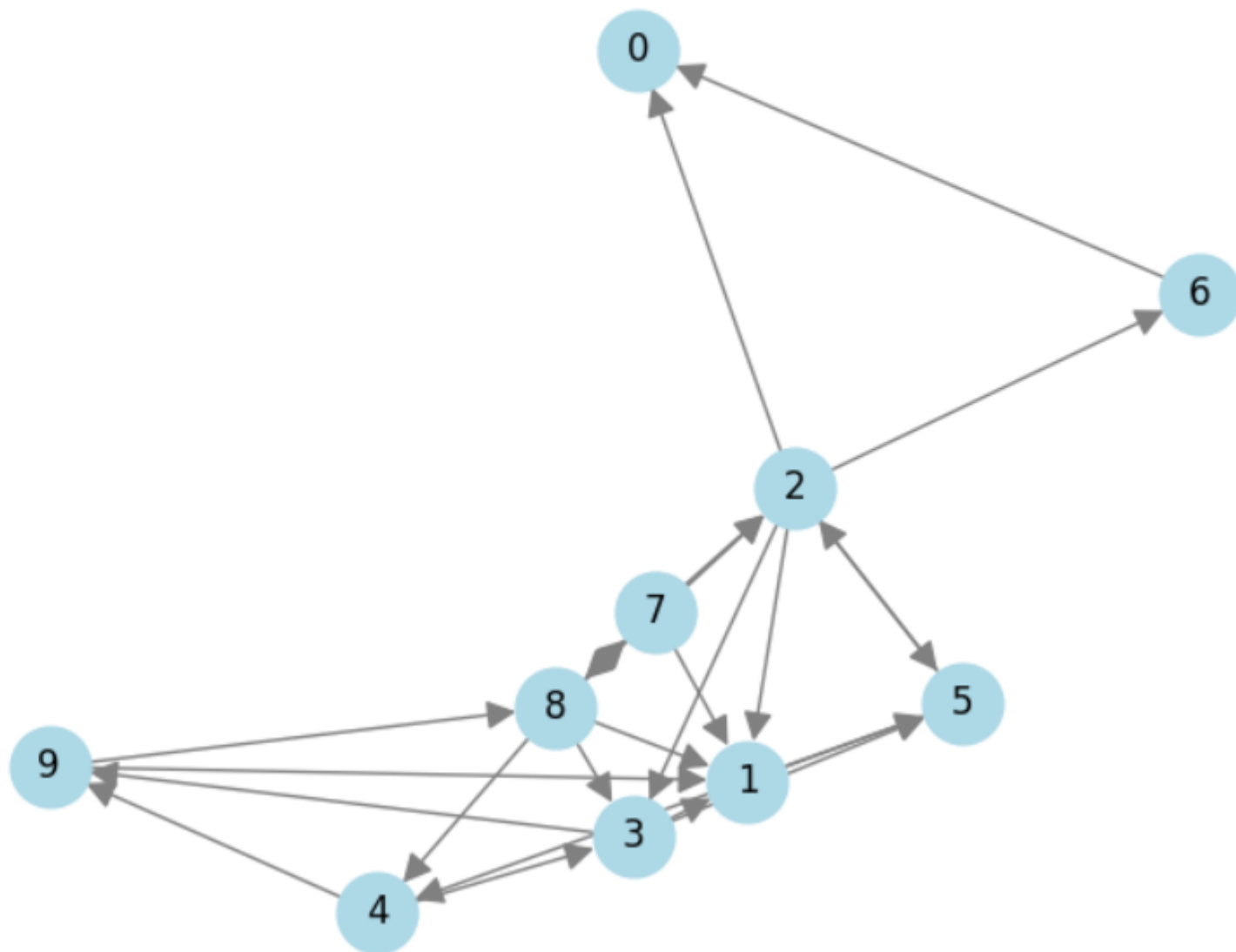


640x480 PNG (32-bit color) 28,05 kB

```

7      class GraphVisualizer:
47          def measure_and_plot_time(self, sizes, matrix_time, list_time):
81              # Побудова графіка часу
82              plt.plot(*args: sizes, matrix_time)
83              plt.plot(*args: sizes, list_time)
84              plt.xlabel("Кількість вершин (n)")
85              plt.ylabel("Час виконання (сек)")
86              plt.title(f"Аналіз часу побудови графа")
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, probability)
97          print("Edges:", builder.get_edges())
98

```



```
class GraphVisualizer:
```

```
    def measure_and_plot_time(self, si
```

```
        # Побудова графіка часу
```

```
        plt.plot(*args: sizes, matrix_t
```

```
        plt.plot(*args: sizes, list_tim
```

```
        plt.xlabel("Кількість вершин (
```

```
        plt.ylabel("Час виконання (сек
```

```
        plt.title(f"Аналіз часу побудо
```

```
        plt.legend()
```

```
        plt.grid(True)
```

```
        plt.show()
```

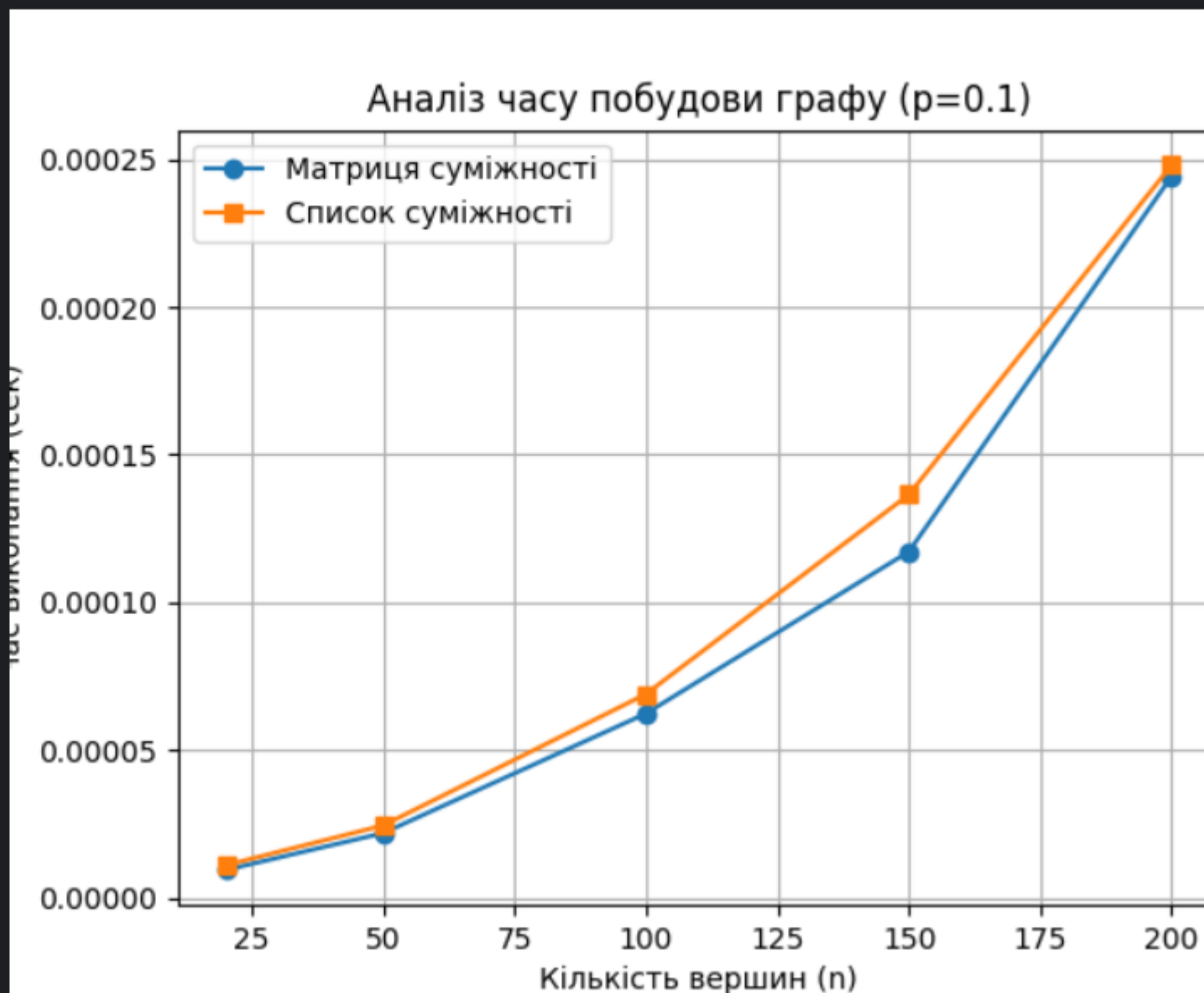
```
num_nodes = 10
```

```
probability = 0.1
```

```
# Створюємо граф для демонстрації
```

```
builder = GraphBuilder(num_nodes, prob
```

```
print("Edges:", builder.get_edges())
```



2

×