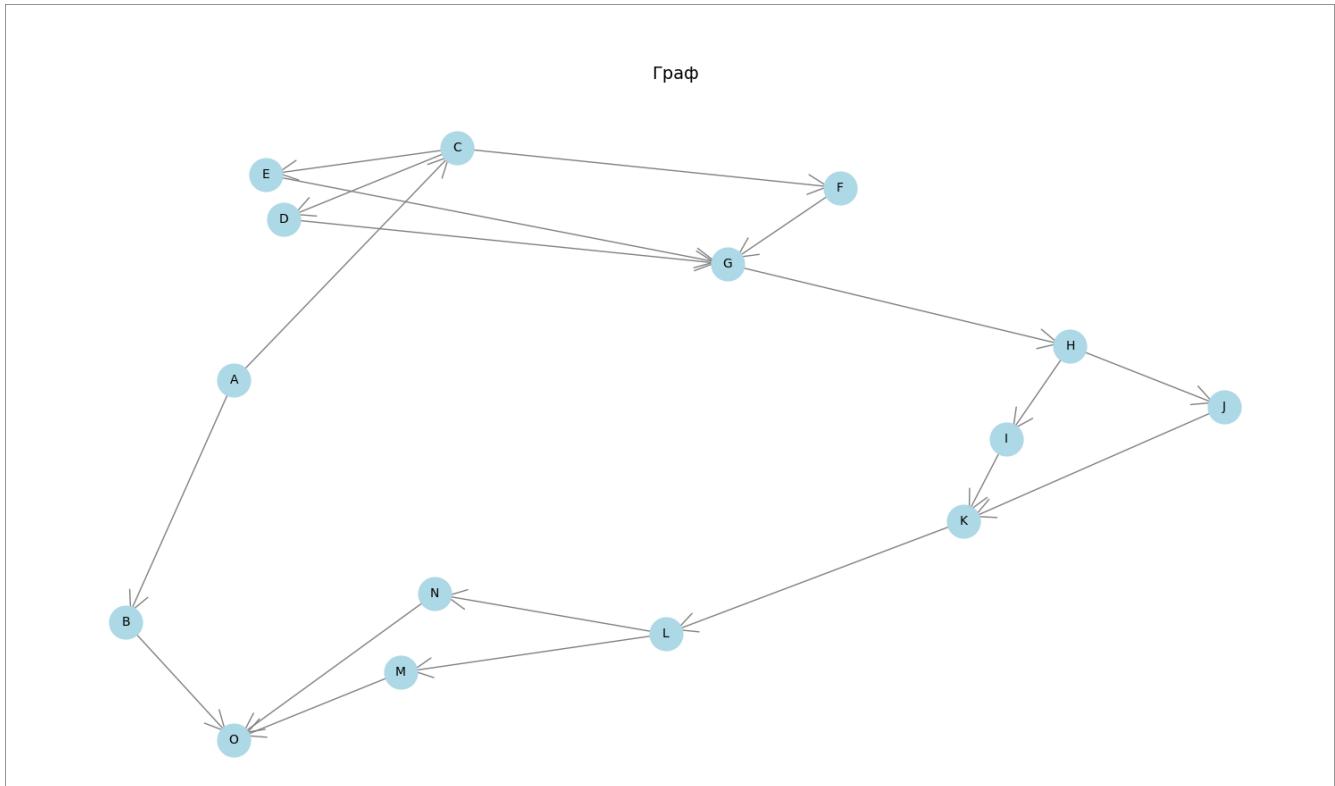


Задача 1. За какое время проект будет выполнен с вероятностью 0,98? Выбор оценок времени выполнения осуществляется по варианту, согласно таблице 19. В таблице: Проц. – процесс, Пред. проц. – предшествующий процесс, П – пессимистическая оценка, О – оптимистическая оценка.

Таблица 1 – Варианты выбора оценок времени

Проц.	Пред. проц.	1	
		П	О
A	-	9	13
B	A	5	11
C	A	10	17
D	C	8	15
E	C	9	16
F	C	9	17
G	D, E, F	10	17
H	G	8	13
I	H	9	17
J	H	10	16
K	I, J	9	18
L	K	5	14
M	L	7	17
N	L	10	18
O	B, M, N	8	13



Граф, построенный программой 1

```
> python h44.py
Промежуточные расчеты (Te и U):
    Проц      Te          U
0      A   11.0  0.444444
1      B    8.0  1.000000
2      C   13.5  1.361111
3      D   11.5  1.361111
4      E   12.5  1.361111
5      F   13.0  1.777778
6      G   13.5  1.361111
7      H   10.5  0.694444
8      I   13.0  1.777778
9      J   13.0  1.000000
10     K   13.5  2.250000
11     L    9.5  2.250000
12     M   12.0  2.777778
13     N   14.0  1.777778
14     O   10.5  0.694444
```

Результаты:

Критический путь: ['A', 'C', 'D', 'G', 'H', 'I', 'K', 'L', 'M', 'O']

Общее критическое время: 118.5

Суммарная дисперсия пути: 14.972222222222223

Нормальное отклонение критического пути: 3.8693955887479667

Время завершения проекта с вероятностью 0.98: 126.43226895693333

Результат выполнения программы 1

Задача 2

По исходным данным, представленным в таблице 20, определить вероятность наступления событий: промежуточных не позже своих поздних сроков. За какой период времени проект будет завершен с вероятностью 0.95?

Таблица 2

Вариант	№ работы	Следует после работы №	Продолжительность		
			Оптимист. оценки	Пессимист. оценки	Вероятн. оценки
1	1	-	1	3	2
	2	-	3	5	3,5
	3	1	2	5	4
	4	2	5	7	5,5

Результат:

Задачи и их данные:

Задача 1: $T_e = 2.00$, $V = 0.11$

Задача 2: $T_e = 3.67$, $V = 0.11$

Задача 3: $T_e = 5.67$, $V = 0.11$

Общее среднее время проекта: 11.33

Общая дисперсия: 0.33

Стандартное отклонение: 0.58

Время завершения проекта с вероятностью 0.95: 12.28

```
> python h4_2.py
Задачи и их данные:
Задача 1: T_e = 2.00, V = 0.11
Задача 2: T_e = 3.67, V = 0.11
Задача 3: T_e = 5.67, V = 0.11

Общее среднее время проекта: 11.33
Общая дисперсия: 0.33
Стандартное отклонение: 0.58
Время завершения проекта с вероятностью 0.95: 12.28
```

Код программ:

1.

```
import networkx as nx
import pandas as pd
```

```
data = {
    "Проц": ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O"],
    "Пред. проц.": [
        None,
        "A",
        "A",
        "C",
        "C",
        "C",
        "D,E,F",
        "G",
        "H",
        "H",
        "I,J",
        "K",
        "L",
        "L",
        "B,M,N",
    ],
    "O": [9, 5, 10, 8, 9, 9, 10, 8, 9, 10, 9, 5, 7, 10, 8],
    "P": [13, 11, 17, 15, 16, 17, 17, 13, 17, 16, 18, 14, 17, 18, 13],
}
```

```
df = pd.DataFrame(data)
```

```
df["Te"] = (df["O"] + 4 * ((df["O"] + df["P"]) / 2) + df["P"]) / 6
df["V"] = ((df["P"] - df["O"]) / 6) ** 2
print("Промежуточные расчеты (Te и V):\n", df[["Проц", "Te", "V"]])
```

```
G = nx.DiGraph()
```

```
for _, row in df.iterrows():
    G.add_node(row["Проц"], Te=row["Te"], V=row["V"])
```

```
for _, row in df.iterrows():
    if row["Пред. проц."]:
        predecessors = row["Пред. проц."].split(",")
        for pred in predecessors:
            G.add_edge(pred.strip(), row["Проц"])
```

```
critical_path = nx.dag_longest_path(G, weight="Te")
critical_time = sum(G.nodes[node]["Te"] for node in critical_path)
critical_variance = sum(G.nodes[node]["V"] for node in critical_path)
critical_std_dev = critical_variance**0.5

Z = 2.05
time_98 = critical_time + Z * critical_std_dev

print("\nРезультаты:")
print(
    f"Критический путь: {critical_path}\n"
    f"Общее критическое время: {critical_time}\n"
    f"Суммарная дисперсия пути: {critical_variance}\n"
    f"Нормальное отклонение критического пути: {critical_std_dev}\n"
    f"Время завершения проекта с вероятностью 0.98: {time_98}"
)
```

2.

```
import numpy as np

tasks = {
    1: {"O": 1, "P": 3, "M": 2},
    2: {"O": 3, "P": 5, "M": 3.5},
    3: {"O": 5, "P": 7, "M": 5.5},
}

for task_id, task in tasks.items():
    O = task["O"]
    P = task["P"]
    M = task["M"]

    T_e = (O + 4 * M + P) / 6

    V = ((P - O) / 6) ** 2

    task["T_e"] = T_e
    task["V"] = V

total_time = sum(task["T_e"] for task in tasks.values())
total_variance = sum(task["V"] for task in tasks.values())

std_dev = np.sqrt(total_variance)

Z = 1.645
time_95 = total_time + Z * std_dev

print("Задачи и их данные:")
for task_id, task in tasks.items():
    print(f"Задача {task_id}: T_e = {task['T_e']:.2f}, V = {task['V']:.2f}")

print(f"\nОбщее среднее время проекта: {total_time:.2f}")
print(f"Общая дисперсия: {total_variance:.2f}")
print(f"Стандартное отклонение: {std_dev:.2f}")
print(f"Время завершения проекта с вероятностью 0.95: {time_95:.2f}")
```