Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ПРАКТИЧЕСКОЙ РАБОТЕ №6
дисциплины «Алгоритмизация»

Выполнил:
Мотовилов Вадим Борисович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Информатика и вычислительная
техника», очная форма обучения

_____
(подпись)

Руководитель практики:
Воронкин Роман Александрович
_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2023 г.

Порядок выполнения работы:

1. Написал программу по задаче покрытия точек отрезками единичной длины двумя способами: обычным (pointscover1) и улучшенным (pointscover2) алгоритмами.

proalg1.py > pointscover1

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import random as rnd
import matplotlib.pyplot as plt


def plot_segments(point, s):
    plt.xlabel('Координаты')
    plt.title("Графическое представление")
    for i in s:
        d = [0, 0]
        plt.plot(i, d, color="blue", linewidth=40, solid_capstyle='butt')
    plt.plot(point, [0 for _ in range(len(point))], linestyle='None',
             marker='|', markersize=60, color="red")
    ax = plt.gca()
    ax.set_yticks([])
    plt.show()

def pointscover1(s):
    segment = []
    while (len(s) > 0):
        xm = min(s)
        segment.append([xm, xm+1])
        i = 0
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\1\algoritm6\prog> & "C:/Program Files/Python311/python.exe" c:/Users/1/algoritm6/prog/proalg1.py
Множество точек: [8.3, 4.8, 8.7, 7.6, 0.7, 7.8, 4.2, 5.7, 8.8, 9.3, 6.2, 3.5, 9.2, 2.9, 9.6, 6.0, 0.4, 2.8, 3.0, 3.6]
Множество отрезков 1: [[0.4, 1.4], [2.8, 3.8], [4.2, 5.2], [5.7, 6.7], [7.6, 8.6], [8.7, 9.7]]
Множество отрезков 2: [[0.4, 1.4], [2.8, 3.8], [4.2, 5.2], [5.7, 6.7], [7.6, 8.6], [8.7, 9.7]]
Минимальное количество отрезков,     которыми можно покрыть данное множество точек =  6
```

Рисунок 1. Код и вывод кода в терминал

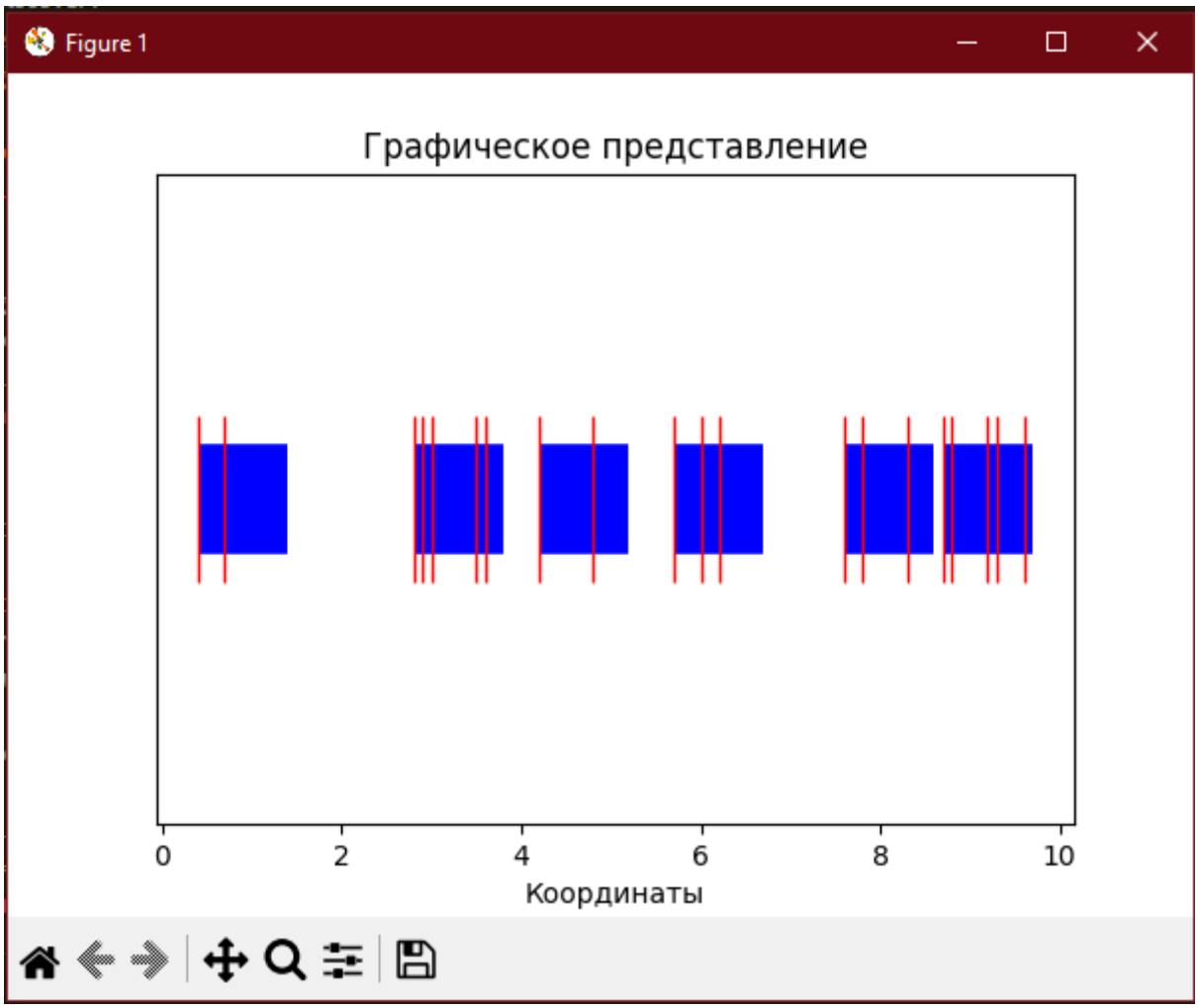

Рисунок 2. Результат работы программы main

2. Написал программу по задаче о выборе заявок, в которой требуется найти максимальное количество попарно не пересекающихся отрезков двумя способами: обычным (actsel1) и улучшенным (actsel2) алгоритмами.

```
proalg1.py  proalg2.py  proalg3.py  proalg4.py
proalg2.py > ...
1   #!/usr/bin/env python3
2   # -*- coding: utf-8 -*-
3
4   from random import randint
5   import matplotlib.pyplot as plt
6
7   def plot_segments(s, sol):
8       plt.xlabel('Координаты')
9       plt.ylabel('Отрезки')
10      k = 40
11      for i in s:
12          d1 = [k, k]
13          plt.plot(i, d1, color="blue", linewidth=10, solid_capstyle='butt')
14          k -= 10
15      d2 = [50, 50]
16      for k in range(len(sol)):
17          plt.plot(sol[k], d2, color="red", linewidth=10, solid_capstyle='butt')
18      ax = plt.gca()
19      ax.set_yticks([])

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                    Python

PS C:\Users\1\algoritm6\prog> & "C:/Program Files/Python311/python.exe" c:/Users/1/algoritm6/prog/proalg2.py
Массив отрезков:  [[181, 463], [538, 928], [942, 996], [800, 863], [701, 976], [625, 800], [201, 466], [301, 1034], [201, 1005], [728, 1023], [882, 1001], [5
01, 932], [712, 746], [825, 837], [676, 913], [872, 896], [826, 1026], [19, 475], [771, 1048], [25, 1069]]

Непересекающиеся отрезки:  [[181, 463], [712, 746], [825, 837], [872, 896], [942, 996]]

                ------------------------

Массив сортированных отрезков:  [[181, 463], [201, 466], [19, 475], [712, 746], [625, 800], [825, 837], [800, 863], [872, 896], [676, 913], [538, 928], [501,
 932], [701, 976], [942, 996], [882, 1001], [201, 1005], [728, 1023], [826, 1026], [301, 1034], [771, 1048], [25, 1069]]

Непересекающиеся сортированные отрезки:  [[181, 463], [712, 746], [825, 837], [872, 896], [942, 996]]
```

Рисунок 3. Код программы ActSel и вывод в терминал

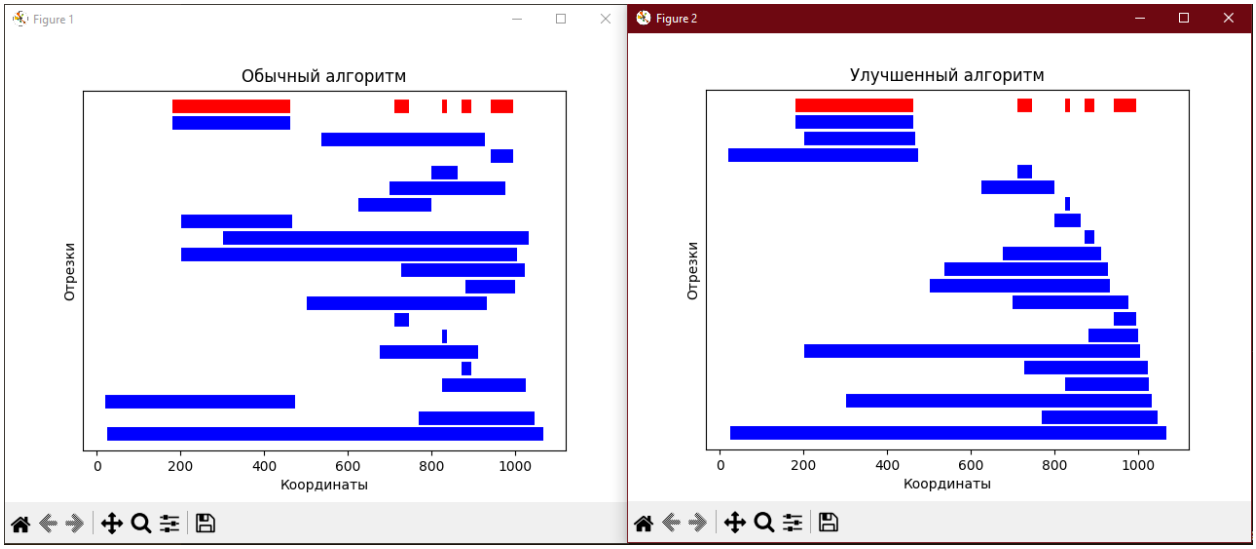

Рисунок 4. Графическое представление решения задачи обоих алгоритмов

3. Написал программу по задаче планирования вечеринки в кампании, в которой требуется по заданному дереву определить независимое множество (множество не соединённых друг с другом вершин) максимального размера.

Рисунок 5. Код программы MaxindSet и вывод в терминал

4. Написал программу по задаче о непрерывном рюкзаке, в которой требуется частями предметов с весами wi, стоимостями ci набрать рюкзак фиксированного размера на максимальную стоимость.

proalg1.py | proalg2.py | proalg3.py | proalg4.py

proalg4.py > ...

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import random
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.patches as patches

def fractional_knapsack(items, capacity):
    items.sort(key=lambda x: x[1]/x[2], reverse=True)
    solution_mas = {}
    solution_money = 0
    total_weight = 0

    for item in items:
        t = capacity - total_weight
        if (t) / item[2] > 1:
            solution_mas[item[0]] = item[2]
            total_weight += item[2]
            solution_money += item[1]
        else:
            solution_mas[item[0]] = t
            solution_money += item[1]/item[2]*t
            break

    return solution_mas, solution_money

if __name__ == '__main__':
    n = 10
    items_mas = [[i, random.randint(1, 100), random.randint(3, 20)]
                 for i in range(n)]
    r = 10
    print("| {:^{r}} | {:^{r}} | {:^{r}} |".format(
```

Рисунок 6. Код программы Knapsack

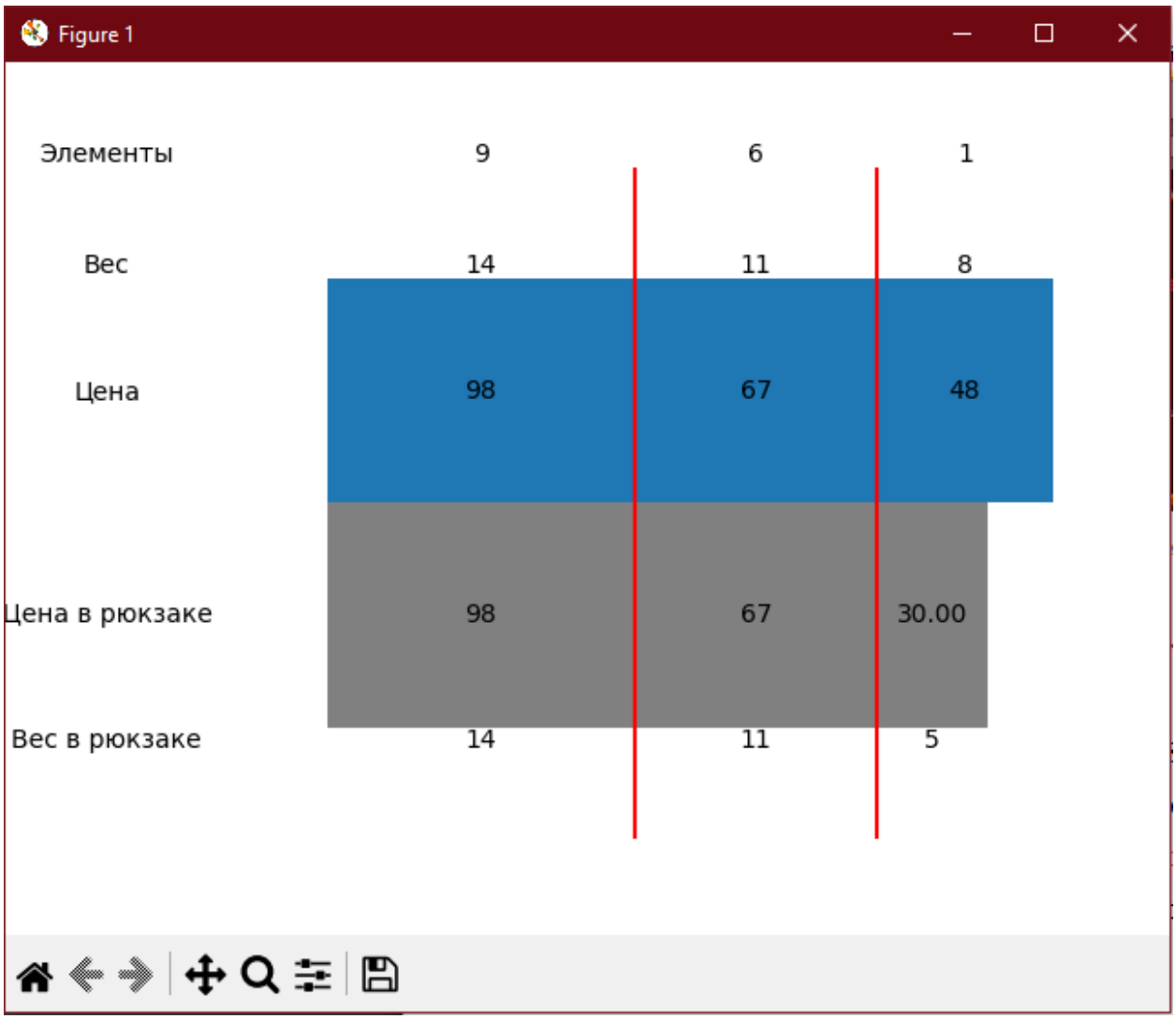

Рисунок 7. Графическое представление решения

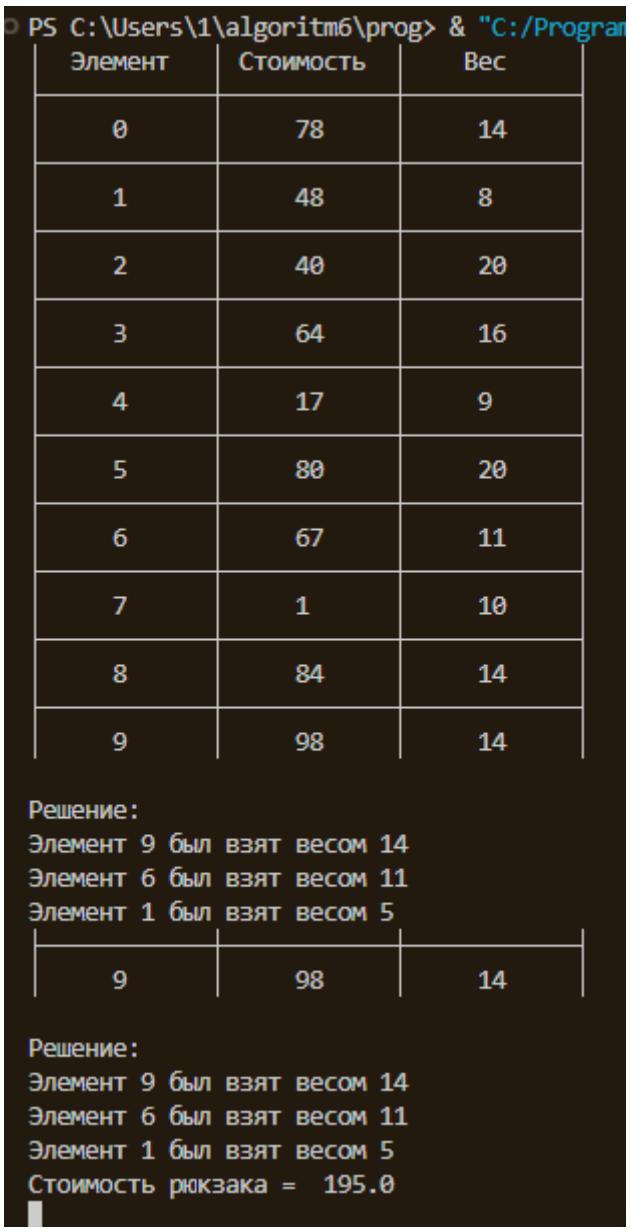

Рисунок 8. Вывод программы Knapsack в консоль

Вывод: в ходе выполнения лабораторной работы были исследованы некоторые из примеров жадных алгоритмов, решающих такие задачи как: задача о покрытии точек минимальным количеством отрезков, задача о нахождении максимального количества попарно непересекающихся отрезков и др.