

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Северо-Кавказский федеральный университет»**

Кафедра инфокоммуникаций

**Отчет по лабораторной работе №6
по дисциплине «Алгоритмизация»**

Выполнил студент группы ИВТ-б-о-22-1

Пушкин Н.С. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2023

Порядок выполнения работы:

Алгоритм находящий минимальные отрезки для покрытия точек:

```
1 import sys
2
3 s = [1, 3, 6, 7, 8, 5]
4 xm = sys.maxsize
5 segment = [[0, 0] for _ in range(len(s))]
6
7 for i in range(len(s)):
8     for ji in range(len(s)):
9         if s[ji] != -1:
10             xm = min(xm, s[ji])
11
12     k = 0
13     segment[i][k] = xm
14     if segment[i][k] == s[i]:
15         s[i] = -1
16
17     k = 1
18     segment[i][k] = xm + 1
19     if segment[i][k] == s[i]:
20         s[i] = -1
21
22     xm = sys.maxsize
23
24 for i in range(len(s)):
25     k = 0
```

Run 1 x

```
C:\Users\Никита\Desktop\alg_lab_6\venv\Scripts\python.exe C:\Users\Никита\Desktop\alg_lab_6\1.py
[1, 2]
[3, 4]
[6, 7]
[7, 8]
[8, 9]
[5, 6]

Process finished with exit code 0
```

Алгоритм находящий минимальные отрезки для покрытия точек с сортировкой:

```
alg_lab_1
venv
1.py
2.py
3.py
External
Scratches

1 import sys
2
3 s = [1, 3, 6, 7, 8, 5]
4 xm = sys.maxsize
5 segment = [[0, 0] for _ in range(len(s))]
6
7 for i in range(len(s)):
8     for ji in range(len(s)):
9         if s[i] != -1:
10             xm = min(xm, s[i])
11
12     k = 0
13     segment[i][k] = xm
14     if segment[i][k] == s[i]:
15         s[i] = -1
16
17     k = 1
18     segment[i][k] = xm + 1
19     if segment[i][k] == s[i]:
20         s[i] = -1
21
22     xm = sys.maxsize
23
24 sorted_segment = sorted(segment, key=lambda x: x[0])
25
26 for i in range(len(s)):
27     k = 0
28     print(f"[{sorted_segment[i][k]}, {sorted_segment[i][k + 1]}]")

Run 2 x
[1, 2]
[3, 4]
[5, 6]
[6, 7]
[7, 8]
[8, 9]
Process finished with exit code 0
```

Улучшенный алгоритм оптимального решения содержащий правый конец отрезка

```

2.py 3.py x 1.py
1 import random
2
3 segments = [(random.randint(a: 1, b: 8), random.randint(a: 2, b: 9)) for _ in range(10)]
4
5 1 usage
6 def print_segments(segments):
7     for segment in segments:
8         print(f"({segment[0]}, {segment[1]})")
9
10 1 usage
11 def initialize_segments():
12     return [(random.randint(a: 1, b: 8), random.randint(a: 2, b: 9)) for _ in range(10)]
13
14 1 usage
15 def main():
16     segments = initialize_segments()
17
18     print("Исходные отрезки:")
19     print_segments(segments)
20
21     segments.sort(key=lambda x: x[1]) # Сортировка по правому концу
22
23     current_segment = segments[0]
24     print(f"\nМинимальный отрезок: ({current_segment[0]}, {current_segment[1]})")
25
26     for i in range(1, len(segments)):
27         if segments[i][0] > current_segment[1]:
28             current_segment = segments[i]
29             print(f"({current_segment[0]}, {current_segment[1]})")
30
31 if __name__ == "__main__":
32     main()

```

Исходные отрезки:

```

(8, 7)
(4, 8)
(6, 2)
(4, 9)
(7, 5)
(2, 4)
(5, 4)
(5, 3)
(8, 8)
(8, 2)

```

Минимальный отрезок: (6, 2)

```

(8, 2)
(5, 3)
(5, 4)
(7, 5)
(8, 7)
(8, 8)

```

Process finished with exit code 0