

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:

Лейс Алексей Вячеславович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем»

(подпись)

Руководитель практики кандидат тех.
наук доцент кафедры
инфокоммуникаций: Воронкин Р.А

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

Порядок выполнения работы:

Алгоритм находящий минимальные отрезки для покрытия точек:

```
cs  [X]
ProgramApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] s = {1,3,6,7,8,5};
            int xm = int.MaxValue;
            int[,] Segment = new int[s.Length, 2];

            int k = 0;
            for (int i = 0; i < s.Length; i++)
            {
                for (int ji = 0; ji < s.Length; ji++)
                {
                    if (s[i] != -1)
                    {
                        xm = Math.Min(xm, s[ji]);
                    }
                }
                k = 0;
                Segment[i, k] = xm;
                if (Segment[i, k] == s[i])
                {
                    s[i] = -1;
                }
                k = 1;
                Segment[i, k] = xm + 1;
                if (Segment[i, k] == s[i])
                {
                    s[i] = -1;
                }
                xm = int.MaxValue;
            }
            for (int i = 0; i < s.Length; i++)
            {
                k = 0;
                Console.Write($"{Segment[i, k]}, ");
                k = 1;
                Console.Write($"{Segment[i, k]}");
                Console.WriteLine();
            }
        }
    }
}
```

```
[1, 2]
[3, 4]
[6, 7]
[7, 8]
[8, 9]
[5, 6]
```

D:\ConsoleApp1\ProgramApp1\bin\Debug\net6.0\ProgramApp1.exe
Работа с консолью
Чтобы вывести текст в консоль, используйте
Сервис "Вывод в консоль" -> "Вывод в консоль"
е отладки
Нажмите на иконку отладки

```
[3, 4]
[7, 8]
[5, 6]
[4, 5]
[1, 2]
[4, 5]
[8, 9]
[5, 6]
[6, 7]
[7, 8]
```

Алгоритм находящий минимальные отрезки для покрытия точек с сортировкой:

```
Program.cs -b X
ConsoleApp2
6
7
8 static void Main(string[] args)
9 {
10     int[] s = new int[10];
11     //сортируем в начале
12
13     for (int i = 0; i < s.Length; i++)
14     {
15         s[i] = new Random().Next(1,9);
16     }
17     Array.Sort(s);
18
19     int xm = int.MaxValue;
20     int[,] Segment = new int[s.Length, 2];
21
22     int k = 0;
23     for (int i = 0; i < s.Length; i++)
24     {
25         for (int ji = 0; ji < s.Length; ji++)
26         {
27             if (s[i] != -1)
28             {
29                 xm = Math.Min(xm, s[ji]);
30             }
31         }
32         k = 0;
33         Segment[i, k] = xm;
34         if (Segment[i, k] == s[i])
35         {
36             s[i] = -1;
37         }
38         k = 1;
39         Segment[i, k] = xm + 1;
40         if (Segment[i, k] == s[i])
41         {
42             s[i] = -1;
43         }
44         xm = int.MaxValue;
45     }
46     for (int i = 0; i < s.Length; i++)
47     {
48
49         k = 0;
50         Console.Write($"{Segment[i, k]}, ");
51         k = 1;
52         Console.Write($"{Segment[i, k]}");
53         Console.WriteLine();
54     }
55 }
56
```

```
[2, 3]
[4, 5]
[5, 6]
[6, 7]
[6, 7]
[7, 8]
[8, 9]
```

```
D:\Conso
кодом 0
Чтобы ав
Сервис"
е отладк
Нажмите
```

```
[2, 3]
[4, 5]
[5, 6]
[6, 7]
[6, 7]
[7, 8]
[8, 9]
```

Алгоритм оптимального решения содержащий правый конец отрезка

```

9      static void Main(string[] args)
10     {
11         InitializeSegments();
12
13         Console.WriteLine("Исходные отрезки:");
14         PrintSegments(segments);
15
16         Array.Sort(segments, (a, b) => a.Item2.CompareTo(b.Item2)); // Сортировка по правому концу
17
18         Tuple<int, int> currentSegment = segments[0];
19         Console.WriteLine($"Минимальный отрезок: ({currentSegment.Item1}, {currentSegment.Item2})");
20
21         for (int i = 1; i < segments.Length; i++)
22         {
23             if (segments[i].Item1 > currentSegment.Item2)
24             {
25                 currentSegment = segments[i];
26                 Console.WriteLine($"({currentSegment.Item1}, {currentSegment.Item2})");
27             }
28         }
29
30         Console.ReadKey();
31     }
32
33     static void InitializeSegments()
34     {
35         Random random = new Random();
36
37         for (int i = 0; i < segments.Length; i++)
38         {
39             int start = random.Next(1, 9);
40             int end = random.Next(start + 1, 10); // Гарантия, что конец больше начала
41             segments[i] = Tuple.Create(start, end);
42         }
43     }
44
45     static void PrintSegments(Tuple<int, int>[] segments)
46     {
47         foreach (var segment in segments)
48             Console.WriteLine($"({segment.Item1}, {segment.Item2})");
49     }
50 }
51

```

Исходные отрезки:
(7, 8)
(1, 5)
(2, 4)
(1, 2)
(1, 6)
(6, 7)
(8, 9)
(4, 5)
(4, 5)
(5, 8)

Минимальный отрезок: (1, 2)
(4, 5)
(6, 7)
(8, 9)

Улучшенный алгоритм оптимального решения содержащий правый конец отрезка

```

1  using System;
2  using System.Linq;
3
4  namespace ConsoleApp3
5  {
6      class Program
7      {
8          static public Tuple<int, int>[]
9
10         static void Main(string[] args)
11         {
12             InitializeSegments();
13
14             Console.WriteLine("Исходные отрезки:");
15             PrintSegments(segments);
16
17             MergeSort(segments, 0, segments.Length - 1);
18
19             Tuple<int, int> currentSegment = segments[0];
20             Console.WriteLine($"Минимальный отрезок: ({currentSegment.Item1}, {currentSegment.Item2})");
21
22             for (int i = 1; i < segments.Length; i++)
23             {
24                 if (segments[i].Item1 > currentSegment.Item2)
25                 {
26                     currentSegment = segments[i];
27                     Console.WriteLine($"({currentSegment.Item1}, {currentSegment.Item2})");
28                 }
29             }
30
31             Console.ReadKey();
32         }
33     }
34 }

```

Исходные отрезки:
(4, 9)
(6, 9)
(1, 5)
(5, 8)
(8, 9)
(1, 6)
(5, 9)
(4, 9)
(6, 8)
(1, 2)

Минимальный отрезок: (1, 2)
(5, 8)