

MINISTRY OF EDUCATION REPUBLIC OF BELARUS
ESTABLISHMENT OF EDUCATION
"BREST STATE TECHNICAL UNIVERSITY"

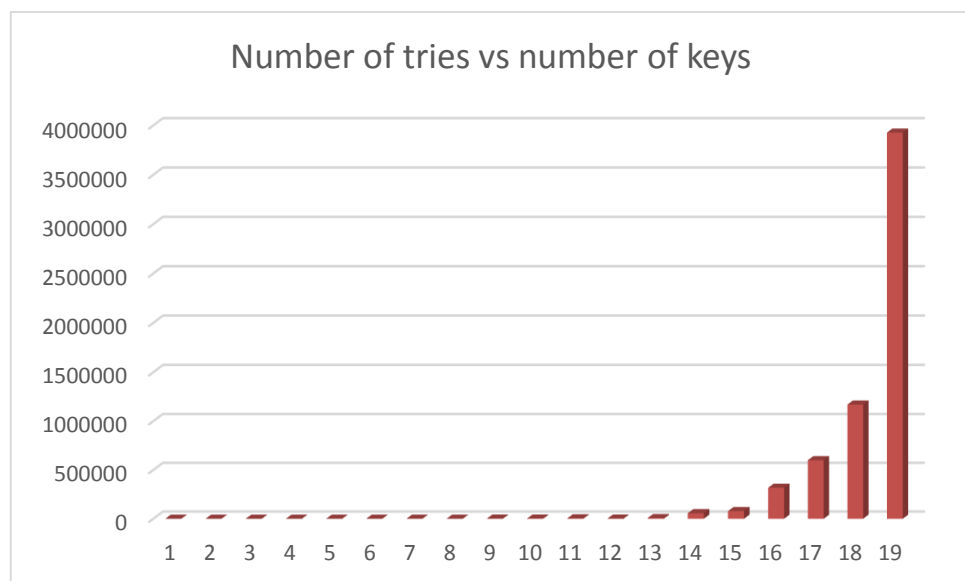
Practice work №5
«Evolutionary Computation»
Subject: «Needle in the haystack»

Made by:
Alexey Cherkasov
Checked by:
Professor Akira Imada

We should generate 19 different binary pin-codes from 2 to 20 numbers and generate random numbers to unlock this pin-codes. We make 10 attempts to every pin-code and take average count of tries, and make graphic with taken results.

Key	Iterations
2	1
3	4
4	1
5	6
6	37
7	16
8	636
9	161
10	1438
11	1550
12	3817
13	1279
14	7253
15	56709
16	78183
17	317958
18	599766
19	1170101
20	3926512

Graphic of number of keys and number of attempts to unlock them



The number of tries exponentially grows

Code program(C#):

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.IO;  
  
namespace pincode
```

```

{
class Program
{
    static void Main(string[] args)
    {
        var average = new List<int>();
        Random rnd = new Random();
        for(int i = 2; i <= 20; i++)
        {

            var test = new List<int>();

            for(int j = 0; j < i; j++)
            {
                test.Add(rnd.Next() % 2);
            }
            for (int z=0; ;z++)
            {
                bool IsOk = true;
                var x = new List<int>();
                for (int j = 0; j < i; j++)
                {
                    x.Add(rnd.Next() % 2);
                }
                for(int j = 0; j < i; j++)
                {
                    if (test[j] != x[j])
                    {
                        IsOk = false;
                        break;
                    }
                }
                if (IsOk)
                {
                    average.Add(z);
                    break;
                }
            }
        }
        StreamWriter wrtite = new StreamWriter("123.txt");
        for(int i = 0; i < average.Count; i++)
        {
            Console.WriteLine(average[i]);
            wrtite.WriteLine(average[i]);
        }
        wrtite.Close();
        Console.ReadKey();
    }
}
}

```