

Needle in a haystack

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace лаба7
{
    public class Class
    {
        List<int> chromosoma = new List<int>();
        List<int> steps = new List<int>();
        Random rnd = new Random();
        public Class()
        {
            //генерируем рандомную двойную хромосому
            chromosoma.Add(rnd.Next(0, 2));
            chromosoma.Add(rnd.Next(0, 2));
            //генерируем шаги
            var item = new List<int>();
            for (int j = 0; j < 10; j++)
            {
                int step = 1;
                while (true)
                {
                    for (int i = 0; i < chromosoma.Count; i++)
                    {
                        item.Add(rnd.Next(0, 2));
                    }
                }
            }
        }
    }
}
```

```

        if (item.SequenceEqual(chromosoma))
        {
            break;
        }
        else
        {
            item = new List<int>();
            step++;
        }
    }
    steps.Add(step);
}
}

```

```

public int GeAverage()
{
    return (int)steps.Average();
}

```

```

public void Generation()
{
    var item = new List<int>();
    chromosoma.Add(rnd.Next(0, 2));
    for (int j = 0; j < 10; j++)
    {
        int step = 1;
        while (true)
        {
            for (int i = 0; i < chromosoma.Count; i++)
            {
                item.Add(rnd.Next(0, 2));
            }
        }
    }
}

```

```

        if (item.SequenceEqual(chromosoma))
        {
            break;
        }
        else
        {
            item = new List<int>();
            step++;
        }
    }
    steps.Add(step);
}
}
}
}
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace лаба7

```

```

{
    class Program
    {
        static void Main(string[] args)
        {
            List<int> average = new List<int>();
            Class cl = new Class();
            int i = 0;
            //после первого шага
            average.Add(cl.GeAverage());
        }
    }
}

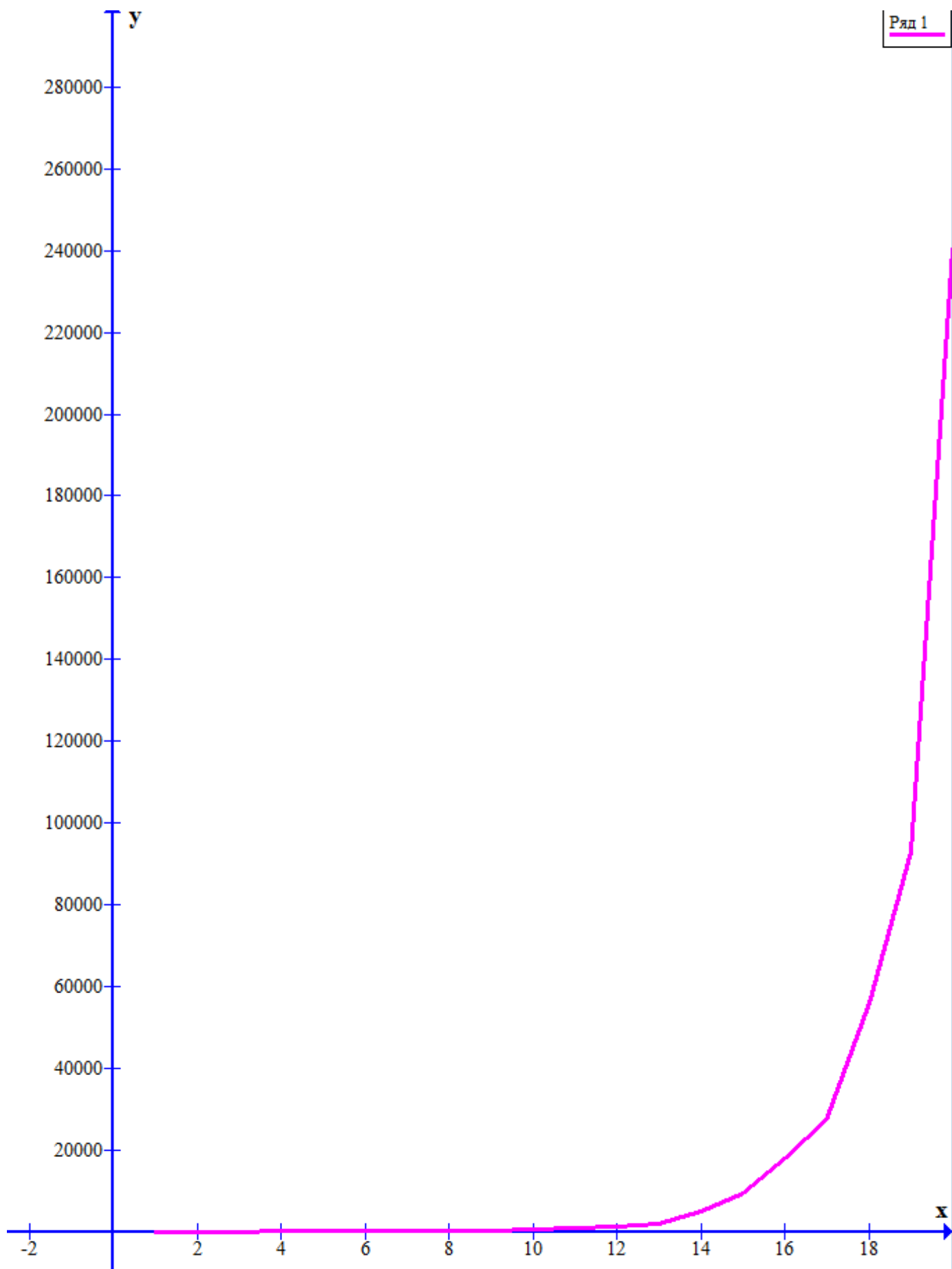
```

```

while(i++ < 19)
{
    cl.Generation();
    average.Add(cl.GeAverage());
    Console.WriteLine(i);
}
for (int j = 0; j < average.Count; j++)
{
    Console.WriteLine(average[j]);
}
}
}

```

1	5
2	7
3	10
4	19
5	43
6	53
7	95
8	149
9	259
10	404
11	881
12	1297
13	2048
14	4929
15	9349
16	17994
17	27274
18	55984
19	92884
20	240426



Название диаграммы

