

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 №2
дисциплины «Алгоритмизация»**

Выполнила:
Кубанова Ксения Олеговна
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А.

(подпись)

Отчет защищен с оценкой_____ Дата защиты_____

Ставрополь, 2023 г.

Порядок выполнения работы

1. Написать рекурсивную программу для нахождения чисел Фибоначчи по алгоритму, данному на лекции. Составить таблицу и график.

Блок-схема:

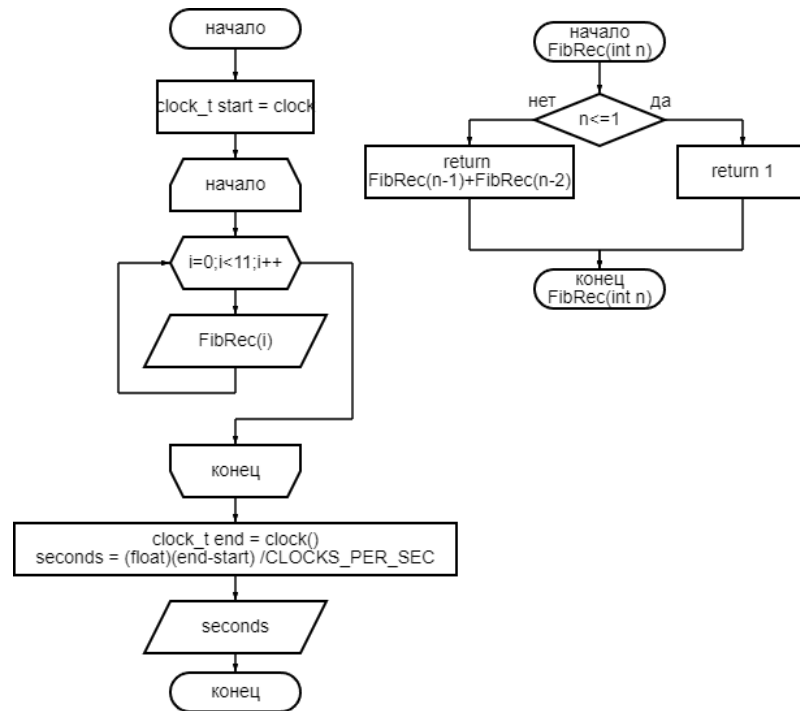


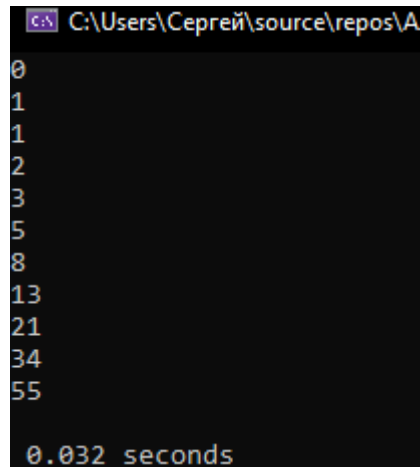
Рисунок 1 - блок-схема к заданию 1

Код:

```
1  #include <ctime>
2  #include <iostream>
3  using namespace std;
4
5  int FibRec(int n)
6  {
7      if (n <= 1)
8      {
9          return n;
10     }
11     else
12     {
13         return FibRec(n - 1) + FibRec(n - 2);
14     }
15 }
16
17 int main()
18 {
19     clock_t start = clock();
20     for (int i=0; i<11; i++)
21     {
22         cout << FibRec(i) << " ";
23     }
24     clock_t end = clock();
25     float seconds = (float)(end - start) / CLOCKS_PER_SEC;
26     cout << " "<<seconds<<" seconds" << endl;
27     system("pause");
28     return 0;
29 }
```

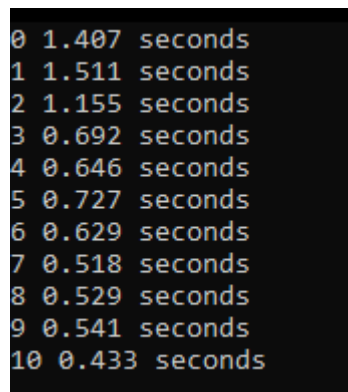
Рисунок 2 - выполнение алгоритма к заданию 1

Результат:



```
C:\Users\Сепрей\source\repos\A
0
1
1
2
3
5
8
13
21
34
55
0.032 seconds
```

Рисунок 3 - результат выполнения алгоритма
и время его выполнения



```
0 1.407 seconds
1 1.511 seconds
2 1.155 seconds
3 0.692 seconds
4 0.646 seconds
5 0.727 seconds
6 0.629 seconds
7 0.518 seconds
8 0.529 seconds
9 0.541 seconds
10 0.433 seconds
```

Рисунок 4 - подсчёт каждого числа при
выполнении 100000 раз

Таблица 1

Алгоритм рекурсии, повторяющийся 100000 раз	
число F	времяT, с
0	1,407
1	1,511
2	1,155
3	0,692
4	0,646
5	0,727
6	0,629
7	0,518
8	0,529
9	0,541
10	0,433
Общее время	8,788

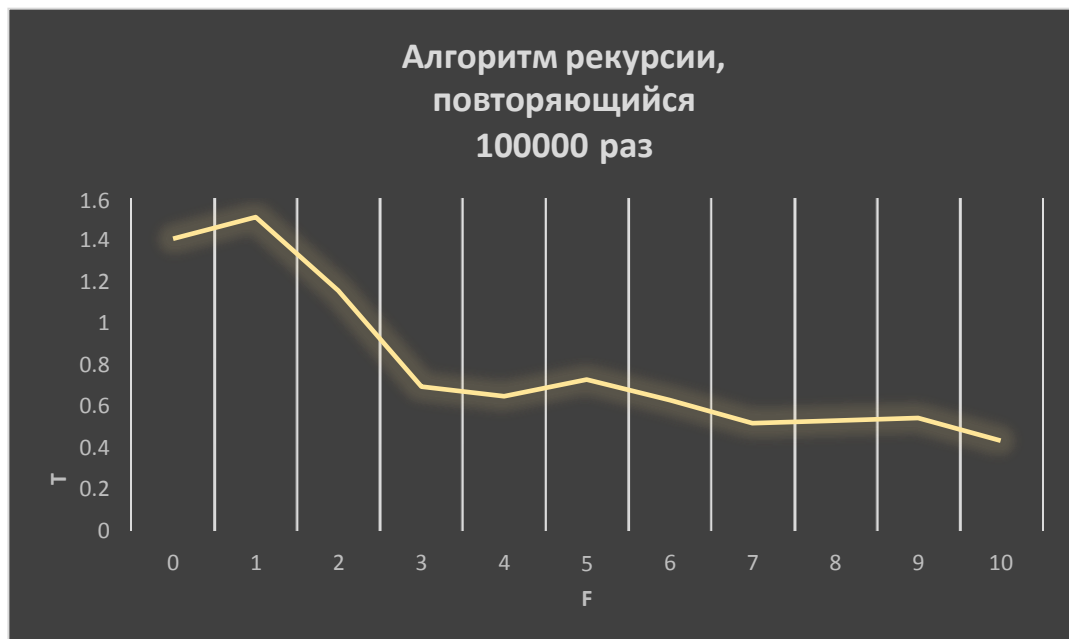


Рисунок 5 – Фибоначчи (рекурсия)

2. Написать программу с массивом, который будет искать число Фибоначчи по алгоритму, данному на лекции. Составить таблица и график.

Блок-схема:

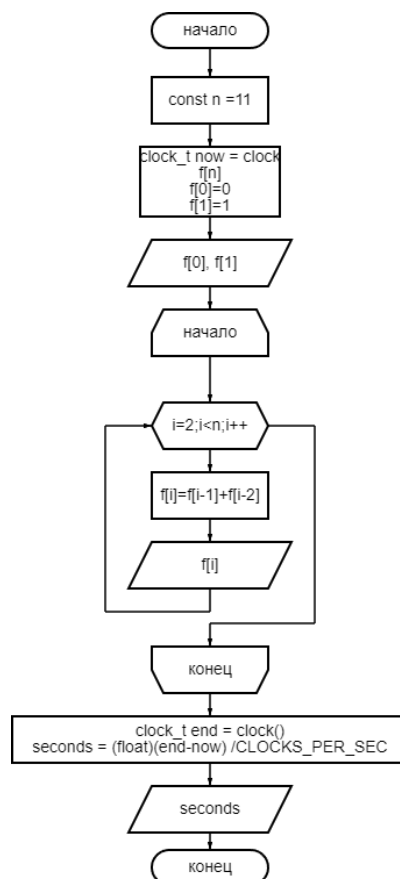


Рисунок 6 - блок-схема к алгоритму 2

Код:

```
1  #include <chrono>
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      int const n=11;
7      clock_t now = clock();
8      double f[n];
9      f[0] = 0;
10     f[1] = 1;
11     cout <<"0."<< f[0] << endl <<"1."<< f[1] << endl;
12     for (int i = 2; i < n; i++)
13     {
14
15         f[i] = f[i - 1] + f[i - 2];
16
17         cout <<i<<"."<< f[i] << endl;
18     }
19     clock_t end = clock();
20     double seconds = (double)(end - now) / CLOCKS_PER_SEC;
21     cout << seconds;
22 }
```

Рисунок 7 - код для алгоритма 2

Результат:

```
0.0
1.1
2.1
3.2
4.3
5.5
6.8
7.13
8.21
9.34
10.55
0.011
```

Рисунок 8 - вывод итога алгоритма и количество секунд, за которое он сработал

```
0. 0.024
1. 0.025
2. 0.026
3. 0.035
4. 0.05
5. 0.078
6. 0.059
7. 0.024
8. 0.024
9. 0.024
10. 0.024
```

Рисунок 9 - подсчёт секунд при 100000 итераций

Таблица 2

Алгоритм массива, повторяющийся 100000 раз	
число F	время T, с
0	0,024
1	0,025
2	0,026
3	0,035
4	0,05
5	0,078
6	0,059
7	0,024
8	0,024
9	0,024
10	0,024
Общее время	0,393

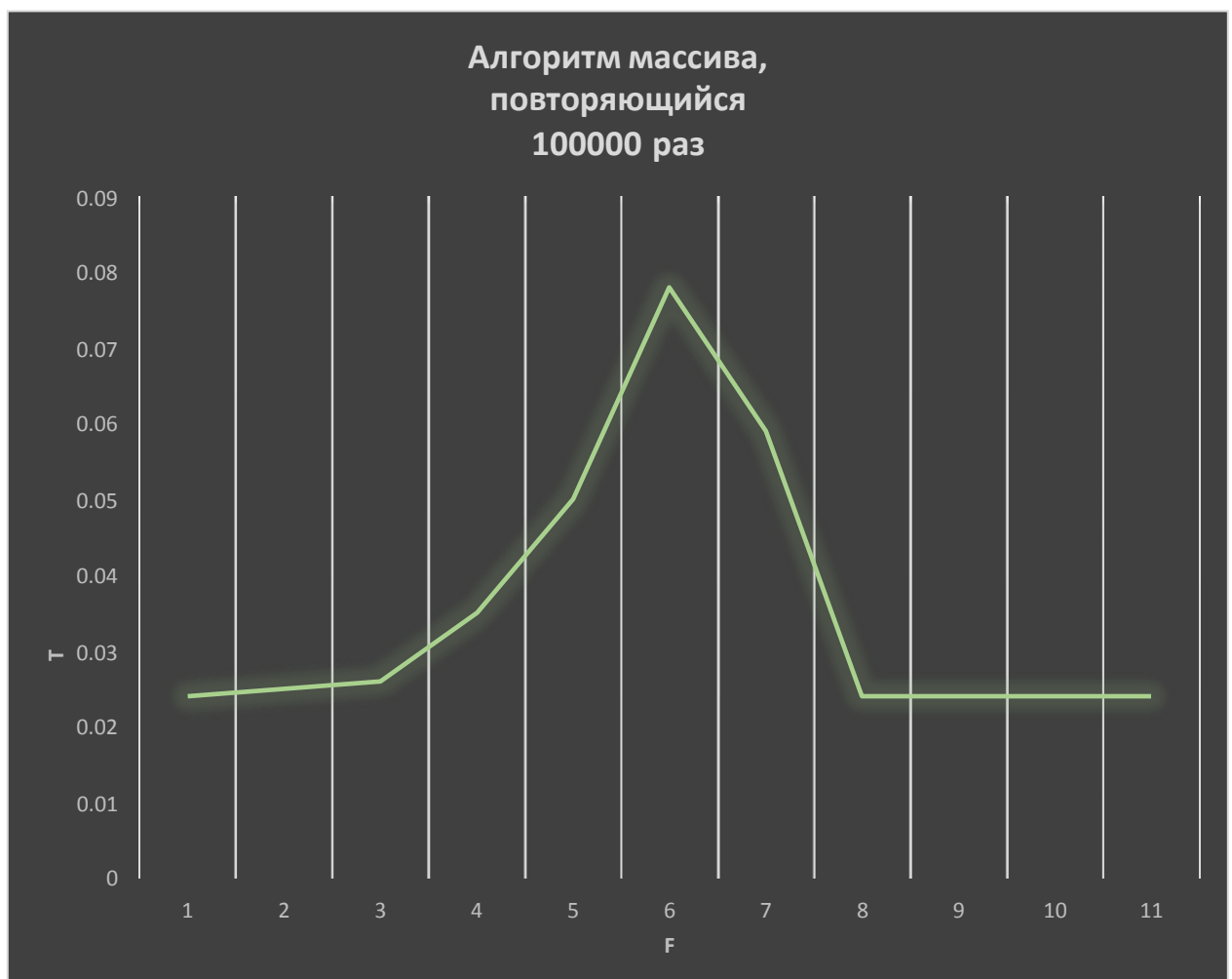


Рисунок 10 – Фибоначчи (массив)

3. Сравнить работу двух алгоритмов.

Таблица 3

Сравнение работы двух алгоритмов		
число F	время T, с, рекурсия	время T, с, массив
0	1,407	0,024
1	1,511	0,025
2	1,155	0,026
3	0,692	0,035
4	0,646	0,05
5	0,727	0,078
6	0,629	0,059
7	0,518	0,024
8	0,529	0,024
9	0,541	0,024
10	0,433	0,024
Общее время	8,788	0,393
Среднее время	0,8	0,4

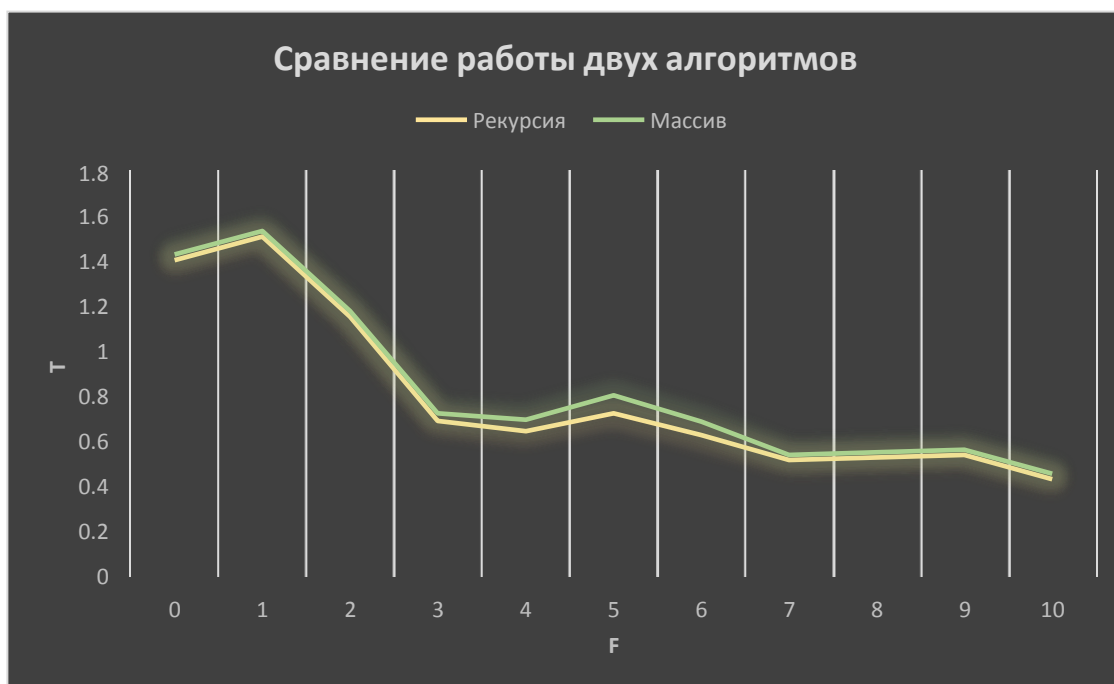


Рисунок 11 – Сравнение алгоритмов

Вывод: в ходе изучения лабораторной работы и построения разного типа алгоритмов и их сравнения можно сделать следующий итог: наивные алгоритмы менее удачны, поскольку на большой объём данных его работы не хватит.