

## Вычисление чисел Фибоначчи

Fib	Рекурсивный $O(2^N)$ алгоритм поиска чисел Фибоначчи.
Fib2	Итеративный $O(N)$ алгоритм поиска чисел Фибоначчи.
Fib3	Алгоритм поиска чисел Фибоначчи по формуле золотого сечения.
Fib4	Алгоритм поиска чисел Фибоначчи $O(\log N)$ через умножение матриц

	Fib	Fib2	Fib3	Fib4
100	-	0	0	0
1_000	-	0	0	0
10_000	-	6	0	0
100_000	-	260	0	46
1_000_000	-	19961	0	4627
10_000_000	-	-	6	459002

## Вычисление колличества простых чисел

Prime	Алгоритм поиска количества простых чисел через перебор делителей, $O(N^2)$
Prime2	Алгоритм поиска простых чисел с оптимизациями поиска и делением только на простые числа, $O(N * \sqrt{N} / \ln(N))$
Prime3	Алгоритм "Решето Эратосфена" для быстрого поиска простых чисел $O(N \log \log N)$
Prime4	Решето Эратосфена с оптимизацией памяти, используя битовую матрицу, сохраняя по 32 значения в одном int, хранить биты только для нечётных чисел.
Prime5	Решето Эратосфена со сложностью $O(n)$
Prime6	Решето Эратосфена, используя битовую матрицу (BitArray класс). Не входит в домашнее задание.

	Prime	Prime2	Prime3	Prime4	Prime5	Prime6
100	0	0	0	0	0	0
1_000	0	0	0	0	1	0
10_000	17	1	0	0	28	0
100_000	1207	5	1	2	2057	1
1_000_000	100482	75	5	22	168207	12
10_000_000	-	1403	67	262	-	129
100_000_000	-	-	1538	2621	-	1412
1_000_000_000	-	-	14218	33131	-	19912

## Вычисление возведения в степень

Power	Итеративный $O(N)$ алгоритм возведения числа в степень.
Power2	Алгоритм возведения в степень через домножение $O(N/2 + \log N) = O(N)$
Power3	Алгоритм возведения в степень через двоичное разложение показателя степени $O(2 \log N) = O(\log N)$

	Power	Power2	Power3
1_000	2	0	0
10_000	2	0	0
100_000	2	0	0
1_000_000	5	1	0
10_000_000	27	3	0
100_000_000	239	77	0
1_000_000_000	2402	1133	2
10_000_000_000	-	-	-

## Выводы:

### Вычисление чисел Фибоначчи

Самый эффективный алгоритм-через умножение Матриц. Самый неэффективный- через рекурсию. Алгоритм через формулу Золотого сечения дает очень не точный результат, из-за округления числа с точкой.

### Вычисление количества простых чисел

Самым эффективным алгоритмом оказался алгоритм решета Эратосфена с хранением битовой матрицы при помощи класса BitArray (Который основывается на массиве int).

### Вычисление возведения в степень

Самым эффективным алгоритмом оказался алгоритм возведения в степень через двоичное разложение показателя степени.

При возведении дробных чисел, во многих случаях не удалось достичь нужной точности. При возведении целых чисел точность была 100%