# Инфраструктура тестирования и обработки результатов

Реализация избранных алгоритмов исполнена на языке C. Также используются некоторые модули Стандартной библиотеки языка Си (libc). Это модуль для измерения времени, модуль для вывода на экран и в файл. Код реализации алгоритмов приводится в Приложении 1. Код функций для фиксирования времени и логирования приводится в Приложении 2.

Порядок тестирования времени исполнения таков. Производится серий измерений времени шифрования. На каждой из серий производится измерений для каждого объема входных данных, . Итого для каждого алгоритма имеется точек, объединенных в групп по кластеров. Результаты тестирования записываются в JSON-файл.

Следующим этапом идет обработка результатов измерения. Она производится на языке средствами библиотеки . Сначала вычисляется значение, полученное по измерениям. Оно считается атомарным, и его погрешность по двум осям вычисляется как случайная погрешность величины, измеренной раз. Это позволяет снизить влияние «промахов» и других случайных факторов. Затем для каждой из серий измерений по атомарных точек строится прямая . Она строится с помощью метода наименьших квадратов. Фиксируются ее значения и и их погрешности. Таким образом фиксируются пар и . Наконец, вычисляется итоговые значения и . Это и будут пропускная способность и задержка алгоритма, соответственно. Код на *Python* приводится в Приложении 3.

# Результаты тестирования

Использовались следующие параметры.

|  |  |
| --- | --- |
| **Параметр** | **Значение** |
| n | 10 |
| p | 100 |
| m | 20 |
|  |  |

Примеры построенных прямых для различных алгоритмов.

Итоговые результаты.

|  |  |  |
| --- | --- | --- |
| **Алгоритм** | **Пропускная способность** | **Задержка** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

В то же время, для данных алгоритмов существуют следующие результаты по времени работы.

|  |  |  |
| --- | --- | --- |
| **Алгоритм** | **Пропускная способность** | **Задержка** |
|  |  |  |
|  |  |
|  |  |
|  |  |  |
|  |  |  |
|  |  |  |