**Отчет Галкиной Таисии Олеговны БПИ213**

Общая часть:

Строки генерировались с помощью srand, чтобы значения зависели от времени, а значит были разными всегда. Места для «?» тоже выбирались рандомно.

В каждом алгоритме мы искали последнее вхождение подстроки в исходный текст. Также если мы встречали «?» в подстроке, то считали, что вместо вопроса стоит тот же символ, что и в тексте относительно той позиции что мы сравниваем.

Данные мы измеряли в наносекундах. Происходило 50 запусков каждого алгоритма на одних и тех же данных, а потом находилось среднее время.

Если текст состоял из 10000 символов:

Итог таков, что обычный КМП лучше, чем КМП с уточненными гранями. Также четвертым алгоритмом была выбрана Z-функция. Она на удивление работает дольше всех. Также наивный алгоритм близок к обычному КМП. Возможна причина тому, что мы рандомно выбираем наш подотрезок, а там по теории вероятности наивному алгоритму очень везло всегда(надо было ему в лотерею сыграть)).

Однако есть еще гипотеза, что подотрезки настолько малы относительно исходного текста, что было проще сравнить посимвольно. Также заметно, что в бинарном алфавите на 10000 символов на маленьких отрезках алгоритмы работают дольше, чем на длинных.

Для текста из 4 букв наивный алгоритм работал лучше всех, после шел обычный КМП, потом КМП с уточненными гранями, а после Z-функция. Также стоит заметить, что на больших отрезках КМП с уточненными гранями и Z-функция работали одинаковое количество наносекунд.

Когда исходный текст 100000 символов:

Также заметим, что, когда текст стал 100000 символов, то алгоритмы работали относительно (очень относительно) за одно и тоже время. А еще в случае, когда алфавит состоял из 2 букв, обычное КМП работало лучше всех. Когда алфавит стал из 4 букв, то КМП обычное и наивный алгоритм работали примерно за одно время.

Вот два графика для сравнения:

**Итог:**  
Заметим, что два лучших алгоритма оказались – наивная реализация и обычное КМП. Хуже всего всегда работала Z-функция.