

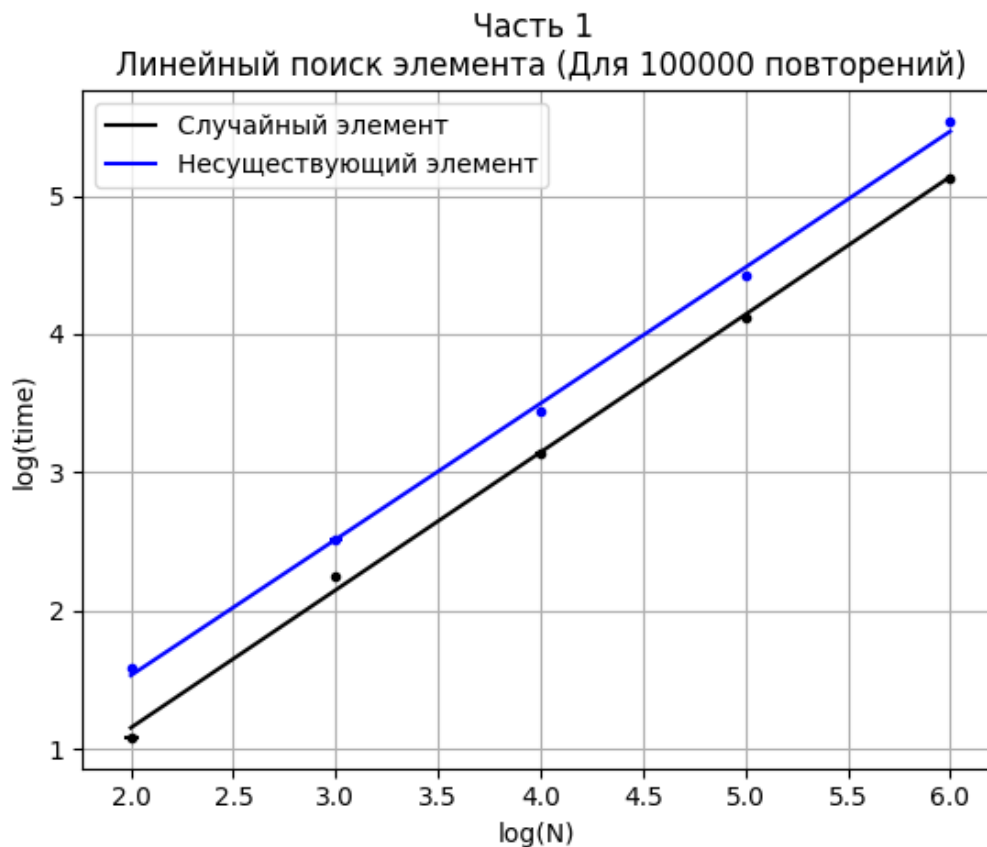
"Лабораторная работа I. Асимптотическая сложность."

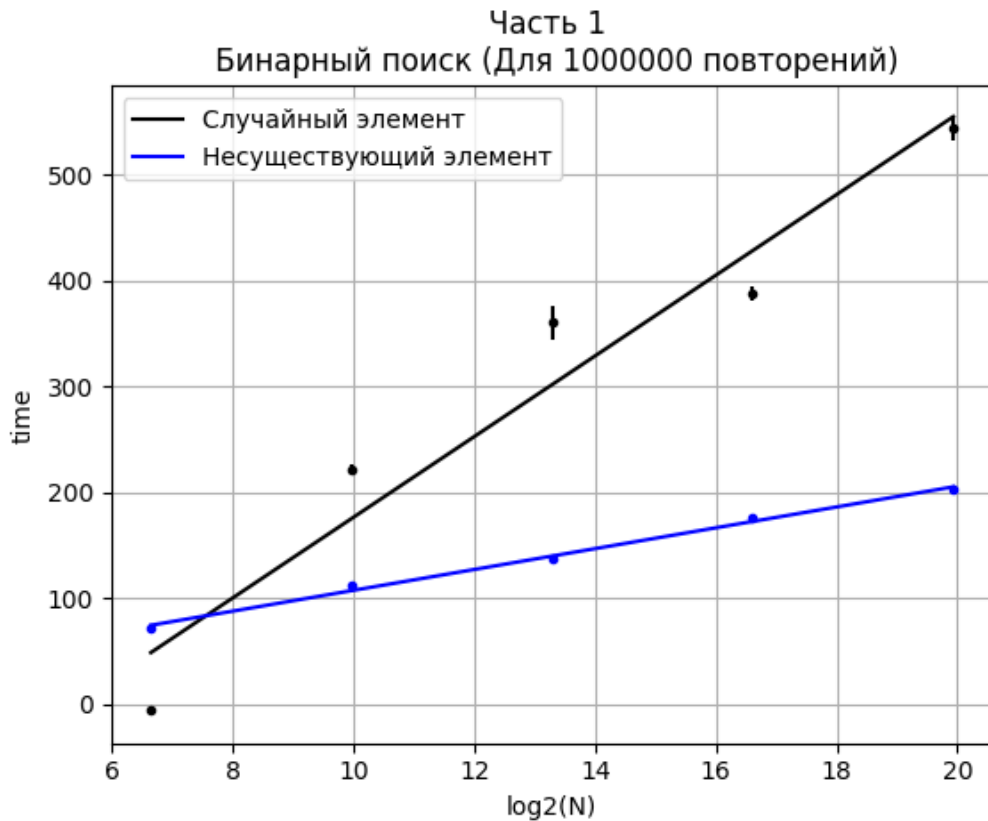
Травина Валерия, Б02-307

23.02.2024г

1 Часть

Здесь все легко. Для начала оценим время выполнения функции $\text{rand()} * \text{rand()} \% N$, поймем что она не зависит от N и в целом временем всех математических операций будем пренебрегать. Только случае использования функции rand() будем вычитать соответствующее время. Для читаемости графиков линейного поиска будем строить их в логарифмических координатах $\log(\text{time})$ от $\log(N)$:





По графикам найдем коэффициенты линеаризации $y = kx + b$:

Таблица 1: Коэффициенты линеаризации

Часть 1. Линейный поиск		
	случайный элемент	несуществующий элемент
k	1.00 ± 0.01	0.98 ± 0.02
b	-0.84 ± 0.03	-0.44 ± 0.02

Часть 1. Бинарный поиск		
	случайный элемент	несуществующий элемент
k	38 ± 4	9.8 ± 0.3
b	204 ± 20	8.8 ± 1.5

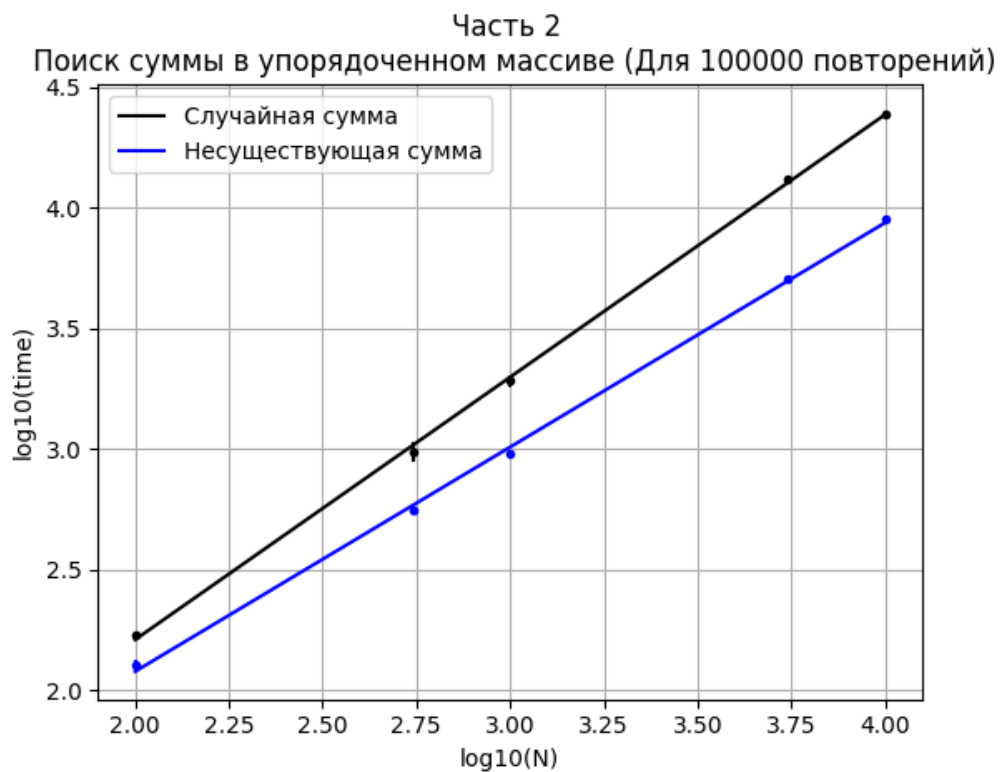
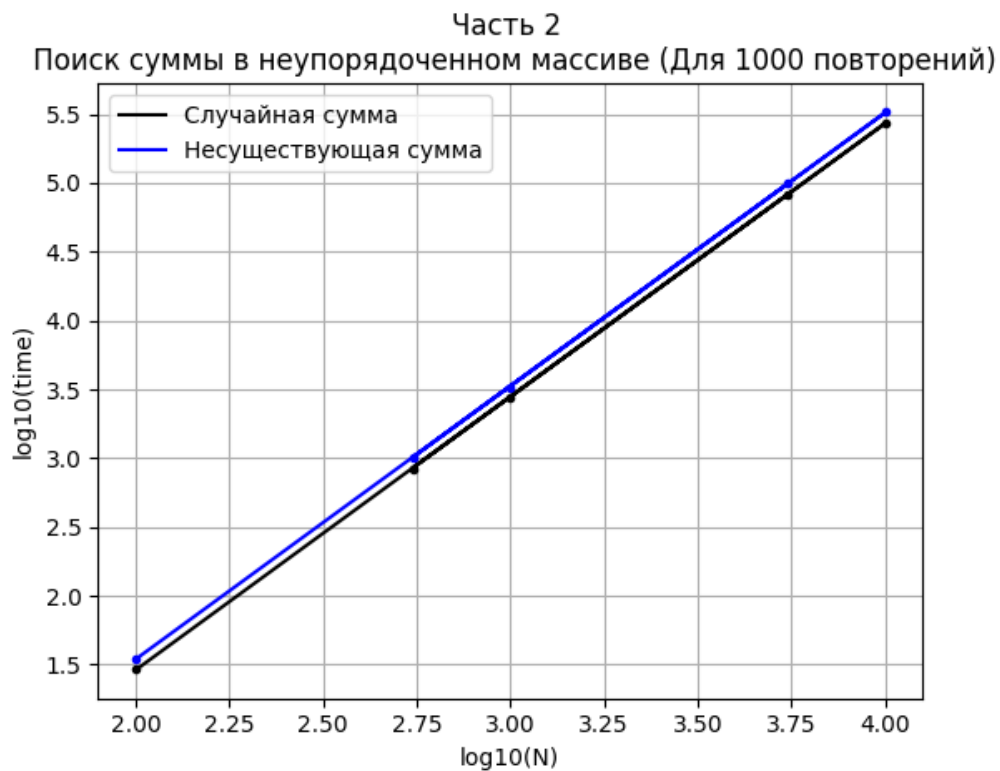
Как и следовало ожидать при линейном поиске время прямопропорционально количеству элементов. При этом время поиска несуществующего элемента (перебор всего массива) в $0.84/0.44 \approx 2$ раза больше времени поиска случайного элемента (так как по теории вероятностей это средний элемент, а зависимость времени от количества элементов линейная).

Результаты бинарного поиска поразительные не только из-за их скорости (сильно большей скорости линейного поиска), но и из-за того, что поиск случайного числа занимает больше времени (даже при учете среднего времени выполнения функции $\text{rand()} * \text{rand()} \approx 200\text{мс}$) нежели поиск несуществующего (т.е перебор всех вариантов). Никакому логическому объяснению этот феномен не поддается (а еще показания ноутбука меняются в 2 раза за пару минут), однако можно заметить интересный факт $38/9.8 \approx 2^2$. Также из результатов очевидно, что $\text{time} \sim \log_2(N)$.

2 Часть

'Трудные времена создают сильных котят'

Построим графики зависимости $\log(\text{time})$ от $(\log(N))$ и найдем коэффициенты линеаризации



Часть 2 Линейный поиск		
	случайный элемент	несуществующий элемент
k	1.987 ± 0.003	1.987 ± 0.004
b	-2.514 ± 0.002	-2.435 ± 0.003

Часть 2 Бинарный поиск		
	случайный элемент	несуществующий элемент
k	1.08 ± 0.01	0.93 ± 0.01
b	0.034 ± 0.007	0.219 ± 0.009

Проанализируем полученные результаты. Во-первых для линейного поиска очевидна степенная зависимость $time \sim \alpha N^2$. Во-вторых из теоретических соображений коэффициенты пропорциональности α должны различаться в 4 раза, однако они почти равны, это может быть следствием оптимизации кода при поиске одного и того же элемента. Рассмотрим оптимизированный поиск в упорядоченном массиве. Зависимость $time \sim N$ линейна.

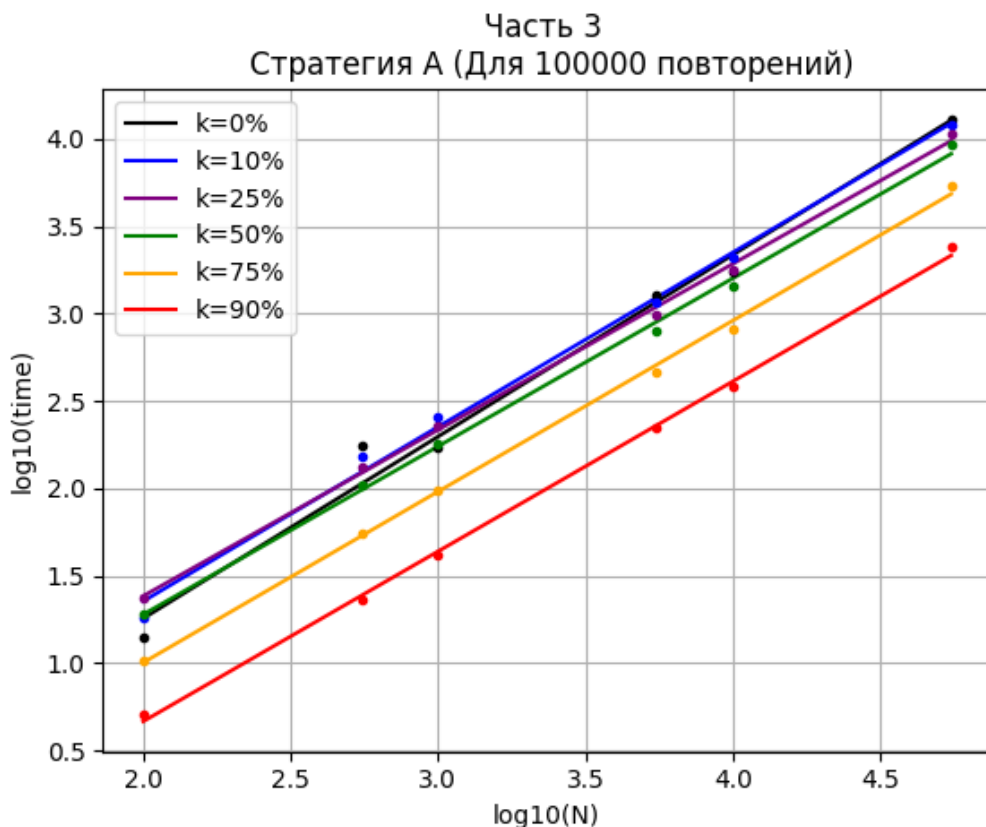
3 Часть

А вот эта уже интересная часть. Для оценки количества повторяющихся элементов введем коэффициент

$$p = \frac{\text{количество_неуникальных_элементов}}{\text{количество_всех_элементов}} \quad (1)$$

Стратегия А

Аналогично предыдущим пунктам построим графики зависимости $\log(time)$ от $(\log(N))$ и найдем коэффициенты линеаризации



Часть 3 Стратегия А			
p (%)	k	b	α
0	1.04 ± 0.05	-0.82 ± 0.04	0,151
10	1.00 ± 0.02	-0.64 ± 0.02	0,229
25	0.95 ± 0.01	-0.52 ± 0.02	0,302
50	0.96 ± 0.02	-0.64 ± 0.02	0,229
75	0.97 ± 0.02	-0.95 ± 0.01	0,112
90	0.97 ± 0.01	-1.28 ± 0.01	0,052

Заметим, что коэффициент наклона почти не меняется и примерно равен 1, поэтому вычислительная сложность такого алгоритма $O(N)$. Однако исследуем зависимость $b(p)$. Пусть $time = \alpha \cdot N^k$, то есть $10^b = \alpha$. Построим графики зависимостей $\alpha(p)$ и $b(p)$:

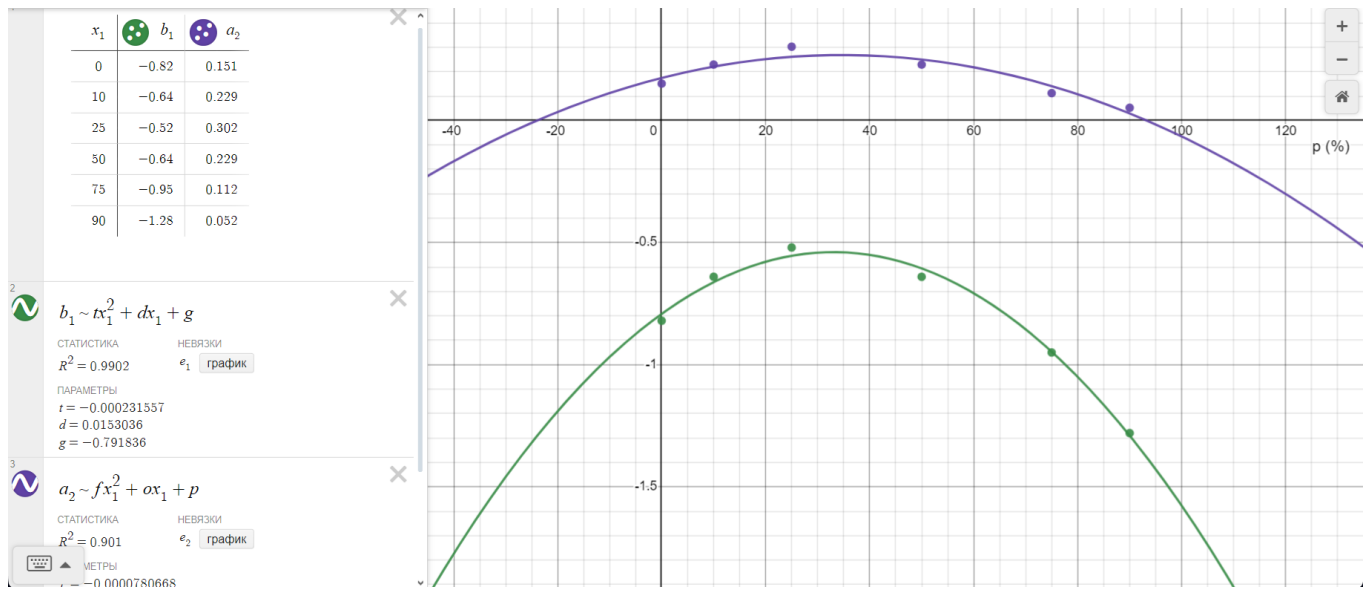
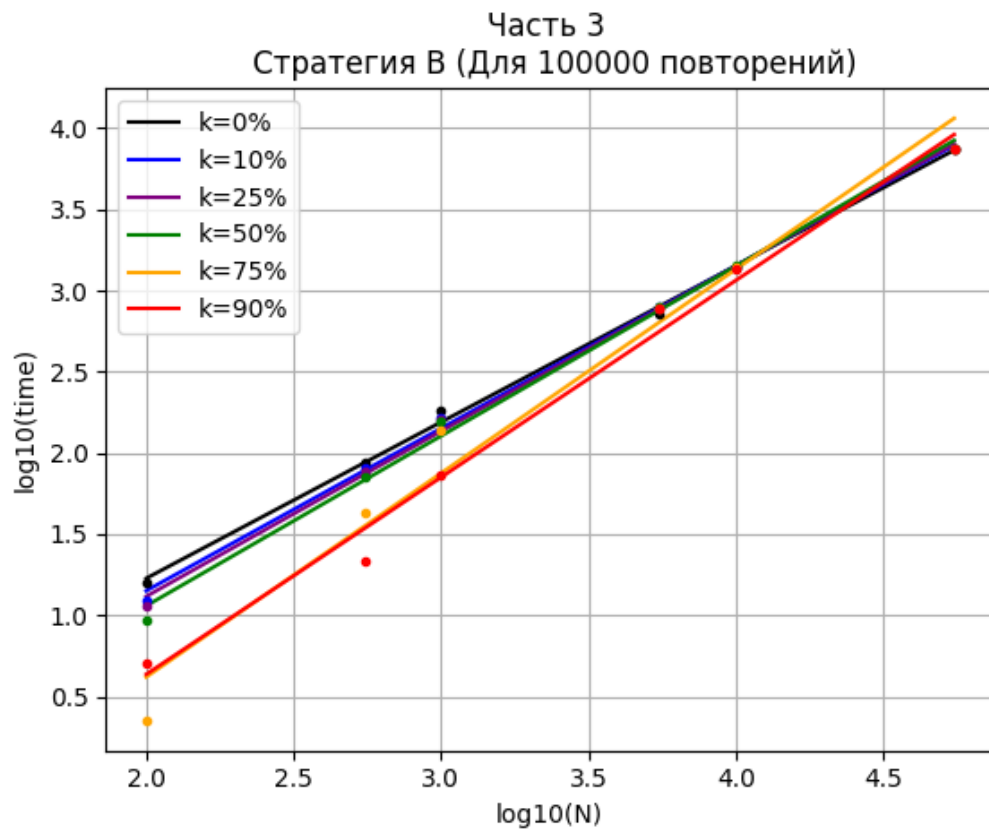


Рис. 1: График зависимостей $\alpha(p)$ -фиолетовый и $b(p)$ -зеленый

Они с приемлимой точностью описываются параболами, что позволяет считать такую стратегию немного более эффективной лишь при $p > 70\%$.

Стратегия В

Опять же построим графики зависимости $\log(time)$ от $(\log(N))$ и найдем коэффициенты линеаризации



Часть 3 Стратегия В			
p (%)	k	b	α
0	0.96 ± 0.02	-0.70 ± 0.01	0,200
10	1.00 ± 0.01	-0.84 ± 0.02	0,145
25	1.02 ± 0.02	-0.91 ± 0.02	0,123
50	1.05 ± 0.03	-1.02 ± 0.02	0,095
75	1.26 ± 0.08	-1.89 ± 0.07	0,013
90	1.22 ± 0.05	-1.78 ± 0.04	0,017

Опять таки посторим график зависимостей $\alpha(p)$ и $b(p)$:

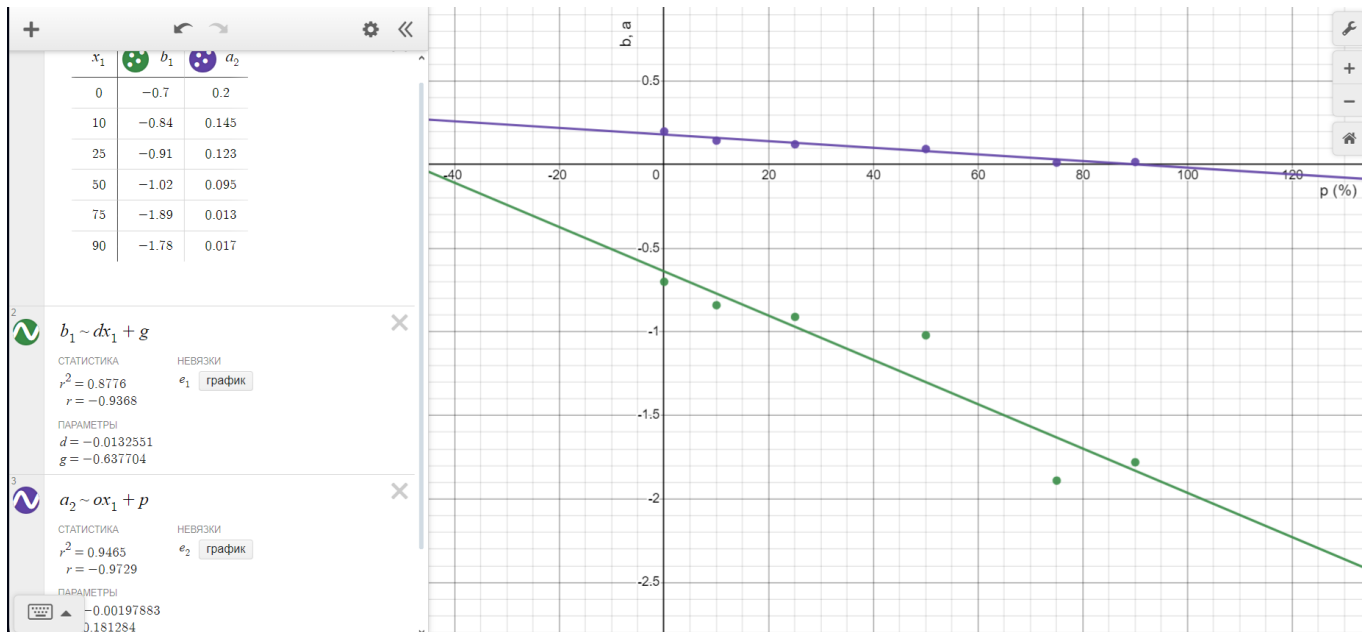
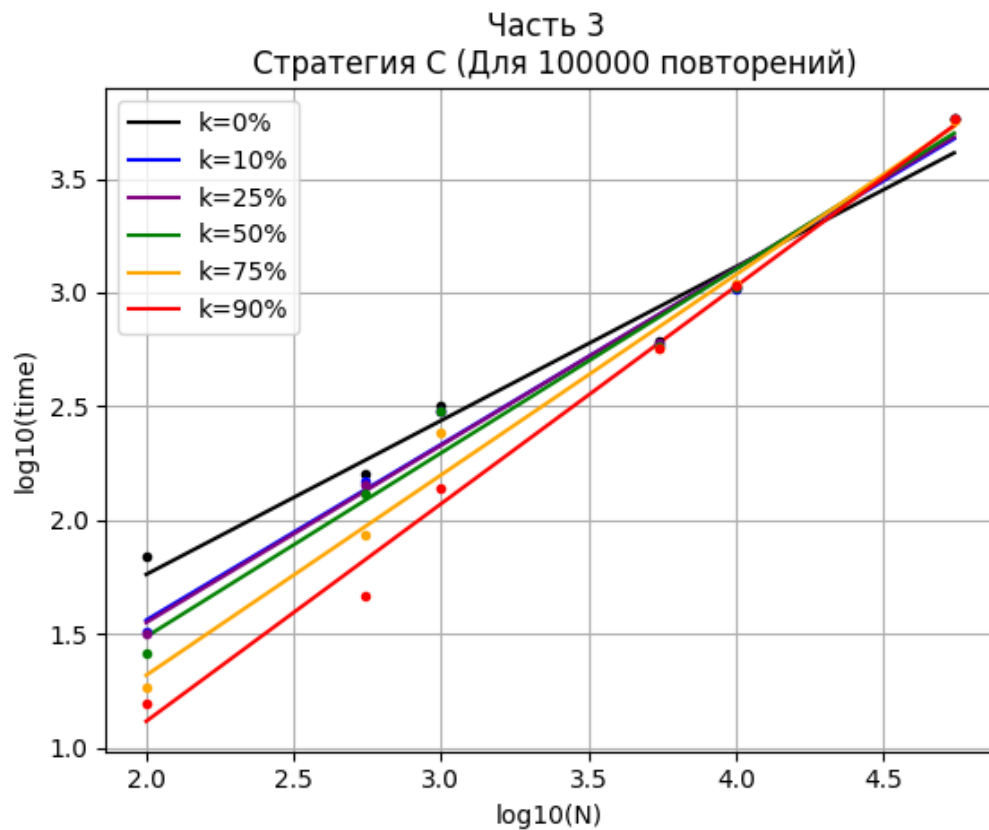


Рис. 2: График зависимостей $\alpha(p)$ -фиолетовый и $b(p)$ -зеленый

Однако этот график менее информативный, так как меняется не только коэффициент, но и степень N . Но все же можно сделать вывод о лучшей применимости этой стратегии по сравнению с предыдущей, так как для небольших массивов (до $N < 10^4$) при любом p время меньше по сравнению с $p=0\%$. Но конечно максимальная эффективность достигается начиная с 70%.

Стратегия С

Не будем сдаваться и построим еще один график $\log(\text{time})$ от $(\log(N))$ и найдем коэффициенты:



Часть 3 Стратегия С			
p (%)	k	b	α
0	$0,68 \pm 0.04$	0.41 ± 0.04	2,57
10	0.77 ± 0.05	0.02 ± 0.04	1,05
25	0.78 ± 0.04	-0.01 ± 0.04	0,98
50	0.81 ± 0.05	-0.13 ± 0.04	0,74
75	0.88 ± 0.04	-0.44 ± 0.04	0,36
90	0.96 ± 0.04	-0.80 ± 0.03	0,17

Строить график зависимостей $\alpha(p)$ и $b(p)$ не будем так как он не информативен, в следствии сильного изменения степени k . Однако по уже построенному графику можно сделать вывод, что для небольших массивов (до $N < 10^4$) при любом p время меньше по сравнению с $p=0\%$. Но, конечно, максимальная эффективность достигается начиная с 60% .

Вывод

Самой удачной оказалась стратегия С, а неудачной - А, не только по скорости убывания времени с увеличением p , но и просто анализируя "чистое" время выполнения функций (см. таблички в экселе).

4 Вывод

Так как эта лабароторная работа имеет номер, значит будут и другие ...