

**Лабораторна робота №3**  
**з предмету**  
**Теоретико числові алгоритми**

**Роботу виконала:**

Бекешева Анастасія

3-го курсу

групи ФІ-12

**Приймав:**

Якимчук Олексій

# 1 Мета.

Ознайомлення з алгоритмом дискретного логарифмування *index-calculus*. Програмна реалізація цього алгоритму та визначення його переваг, недоліків та особливостей застосування. Практична оцінка складності роботи та порівняння різних реалізацій цього алгоритму.

# 2 Постановка задачі.

Написати програму, що реалізовує алгоритм *index-calculus* для груп типу  $\mathbb{Z}_p^*$ .

# 3 Хід роботи.

## 3.1 План.

1. Створення github repo.
2. Імплементация *index-calculus*.
  - (a) Розпаралелювання генерування рівнянь.
  - (b) Імплементация вирішення СЛР.
3. Підняття Docker.

## 3.2 Проблеми.

Були проблеми з розпаралелюванням, спочатку вибрала один модуль і, що тільки не намагалась з ним зробити, але він всеодно погіршував час роботи лаби. Потім просто обрала інший модуль. Також було весело з розв'язком СЛР. Знайшла модуль *galois*, який в комбінації з *numpy* може вирішувати СЛР, але там були обмеження щодо задання генератору поля, тому довелося шукати щось інше. Вийшло імплементувати модифікацію методу Гаусса.

## 3.3 Розпаралелювання

Я використовувала модуль *threading*. Далі увесь *range* чисел розбивається на частини, кожна з яких обробляється окремим потоком. Кожен потік обробляє свою частину чисел паралельно з іншими потоками. Використовується блокування (*self.lock*), щоб уникнути одночасного доступу кількох потоків до спільного списку. Використовується подія (*self.stopevent*), , *ii*.

## 4 Результати.

В таблиці 1 наведено заміри роботи алгоритму СПГ та перебору для чисел довжини від 3 до 15. Перебір перестав працювати на числах довжини 10. СПГ працювало на всіх, але для числ довжини від 8 для задачі 2 типу час був сильно більшим за час для задачі 1 типу. З рисунку ??, що СПГ сильно ефективніше перебору для задачі 1 типу, а з рисунку ?? та ?? видно, що СПГ справляється задачею типу 2 трошки гірше. З рисунку ?? видно що перебір просто в якийсь момент перестає працювати.

## 5 Висновки.

СПГ загалом (очевидно) працює краще перебору. Перевірялись числа від 3 до 15, адже більші числа допоміжна програма просто не згенерувала, але вирогідно СПГ перестало працювати ефективно приблизно на числах довжини від 20.

## 6 Docker.

```
docker run bekeshevaaa/nta-lab-2:0.1 python3 script.py a b p
```

Number of digits	$\alpha$	$\beta$	$p$	$x$	SPH time	BF time
Type 1	179	97	191	168	0.0002	0.0001
Type 2	2	437	491	226	0.0001	0.0001
Type 1	3086	2576	3617	1039	0.0002	0.0005
Type 2	3786	2919	4259	4085	0.0074	0.0021
Type 1	2417	11288	13627	1222	0.0014	0.0006
Type 2	606	19755	33773	9717	0.0108	0.0051
Type 1	69366	534740	889081	630451	0.0005	0.5202
Type 2	409634	294022	415607	221931	0.0275	0.1607
Type 1	3842476	6675652	8043979	7268042	0.0176	7.1154
Type 2	2742375	376513	4981313	1537567	0.0308	1.4344
Type 1	66830006	51535128	87321277	26090344	0.0047	28.4999
Type 2	3472738	5594295	14855123	4985397	13.2654	4.7348
Type 1	194476600	191132926	347205581	141181818	0.0045	164.3148
Type 2	197183379	197183379	256857593	128428797	0.1301	0.0000
Type 1	4551375215	1573551722	7870537313	7709472907	0.0164	300
Type 2	1372336390	1056512366	8080186871	826129382	27.1821	300
Type 1	14837213830	38662976351	44392159481	39230117359	0.0585	300
Type 2	17435029047	10429220418	31788610771	14754109763	16.2017	300
Type 1	543050355399	417773336744	740746544833	549000079782	0.0409	300
Type 2	258386314621	234508449353	948929348839	787937399626	57.1036	300
Type 1	3442360773292	4130940749100	5478710518453	3118837649776	0.0525	300
Type 2	7043450488041	3693873406607	9671170467619	6800381608693	1.7960	300
Type 1	26132929097130	18931503409489	90043656944741	24641734992724	0.3319	300
Type 2	2503177357728	5198261491908	22956856087093	6030786792897	167.3889	300
Type 1	145534237128898	59459578217324	234746379151573	191232194101023	0.0238	300
Type 2	72939729843574	106261771871845	290051083605971	32003267688517	1.7317	300

Таблиця 1: Заміри часу