1. Министерство науки и высшего образования Российской Федерации
2. Санкт-Петербургский политехнический университет Петра Великого
3. Институт компьютерных наук и кибербезопасности
4. Высшая школа кибербезопасности

**ЛАБОРАТОРНАЯ РАБОТА № 1**

1. «Алгоритм Ленстры – Ленстры - Ловаса и его применение»
2. по дисциплине «Быстрые вычислительные алгоритмы»
3. Выполнил
4. студент гр. 5151001/00201 Устюгов А.А.

<*подпись*>

1. Преподаватель д.т.н. Шенец Н.Н.

<*подпись*>

1. Санкт-Петербург
2. 2024

# Цель работы

Изучение алгоритма Ленстры - Ленстры - Ловаса и его применения к криптоанализу раневых криптосистем

# Задание

1. Написать программу, которая реализует LLL – алгоритм.
2. На простых примерах убедиться в корректности реализации. Проверить результаты работы в известных математических пакетах.
3. Реализовать алгоритм решения аддитивной задачи об укладке ранца на базе LLL – алгоритма. Проанализировать, при каких входных данных алгоритм с наибольшей вероятностью успешно завершается в условиях существования решения.
4. Обосновать полученные результаты

# Ход работы

Для реализации был выбран язык C++. Для удобства реализации и чтения кода была реализована структура математического вектора с переопределенными операциями сложения, вычитания, умножения на скаляр и скалярного произведения. Матрица теперь задается как набор реализованных векторов.

LLL – приведенным базисом называется базис, для которого выполнено:

При построении LLL-базиса используется то, что определитель решетки инвариантен относительно процесса ортогонализации. Поэтому с помощью процесса ортогонализации можно уменьшать норму вектора.

Алгоритм приведения базиса к LLL-базису итеративный. Пусть k векторов уже приведены. Если очередной вектор подходит по условиям LLL-приведенности, то он добавляется к LLL-базису, если вектор не подходит по первой проверки, то происходит операция, называемая reduce, т.е. вектор уменьшается по норме, если не проходит по второму, то происходит операция swap, при котором из LLL-базиса уходит последний вектор, и для текущего вектора снова начинается проверка.

На рисунках 1, 2, 3, 4, 5 приведены результаты тестирования и их проверка в SageMath.

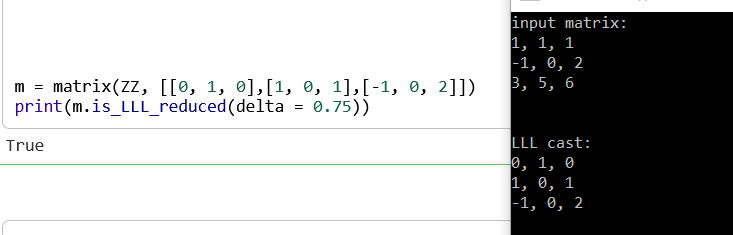


Рисунок 1 – базис из 3 векторов

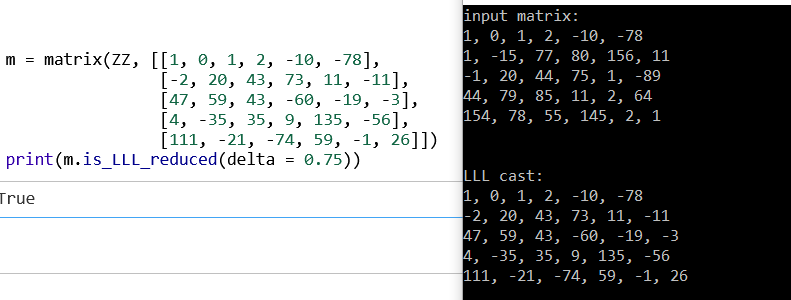


Рисунок 2 – базис из 5 векторов

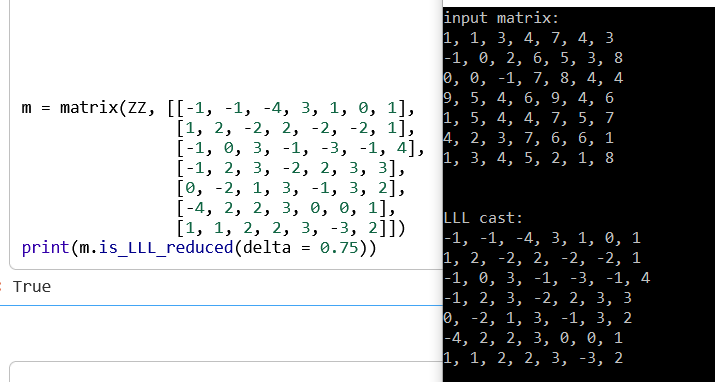


Рисунок 3 – базис из 7 векторов

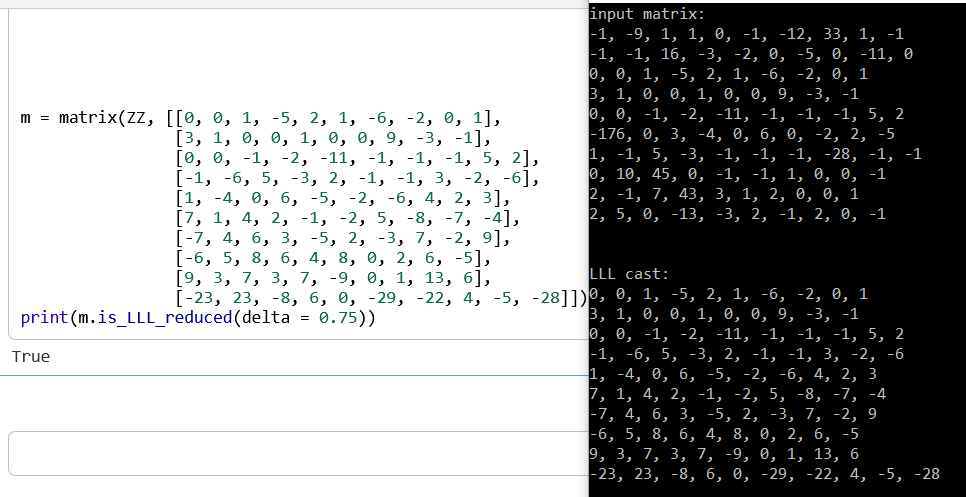


Рисунок 4 – базис из 10 векторов

Далее был реализован алгоритм решения аддитивной задачи об укладке ранца, где основная идея состоит в правильном построении базиса и приведения его к LLL-базису. После получения LLL-базиса, все векторы, чьи коэффициенты по модулю равны 1/2, становятся кандидатами на решение и с их помощью либо восстанавливается укладка, либо решение не найдено.

На рисунке 5 представлено решение укладки ранца из варианта методического пособия.

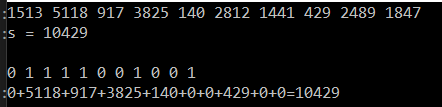


Рисунок 5 – решение задачи об укладке ранца

Для дальнейшего анализа была реализована генерация ключей в ранцевой криптосистеме.

# Контрольные вопросы

1. Является ли LLL-приведенный базис решетки единственным?

Нет.

1. Как можно применить алгоритм решения аддитивной задачи об укладке ранца для решения задачи общего вида (возможно, с неравенствами в условии)?

# Вывод

В ходе выполнения лабораторной работы был изучен LLL-алгоритм и его применение при криптоанализе ранцевой криптосистемы. Данный алгоритм практически всегда восстанавливает укладку ранца с небольшой плотностью. Если плотность больше 2, то вероятность восстановления практически 0.