

# ПРОЕКТ

"Генератор случайных рецептов на основе доступных ингредиентов"

Этот проект — **интеллектуальный помощник для кулинаров**, который подбирает рецепты из базы данных на основе списка продуктов, имеющихся у пользователя.

Главная фишка — **оптимизированный алгоритмический поиск** и **оценка сложности приготовления**.

## Функционал

### а) Ввод данных

- Пользователь вводит список доступных ингредиентов (вручную или загружает из файла)

### б) Поиск рецептов

- Программа анализирует базу данных рецептов и находит те, которые **максимально соответствуют** списку пользователя.
- Критерии поиска
  - Рецепты с наибольшим % совпадения ингредиентов
  - Возможность замены недостающих продуктов (опционально)
  - Фильтрация по сложности (лёгкие/средние/сложные). Можно просто подписать рядом сложность.
- 

### с) Вывод результатов

- Отсортированный список подходящих рецептов с указанием:
  - % совпадения ингредиентов
  - времени приготовления
  - сложности
  - возможных замен

## Пример

Найдено 3 рецепта:

1. Куриный суп (95% совпадения)
  - Не хватает: лавровый лист (можно заменить на сушёный чеснок)
  - Время: 40 мин | Сложность: лёгкая

## 2. Жаркое из курицы (85% совпадения)

- Не хватает: перец (можно исключить)
- Время: 60 мин | Сложность: средняя

# Применяемые алгоритмы и структуры данных

## а) Хранение данных

- База рецептов в формате JSON (или SQLite)

## б) Алгоритмы поиска

- Поиск рецептов с максимальным совпадением ингредиентов
  - Используется хеш-таблица для быстрого подсчёта совпадений
- Алгоритм замены ингредиентов
  - Если у пользователя нет какого-то продукта, программа предлагает замену из графа сочетаемости (например, "сметана → йогурт").
  - Можно реализовать через алгоритм Прима для поиска наиболее близкой замены.
- Оценка сложности приготовления
  - На основе времени и количества шагов
    - Лёгкие: <30 мин, ≤5 ингредиентов.
    - Средние: 30-60 мин, 6-10 ингредиентов.
    - Сложные: >60 мин, многоэтапные рецепты.

## с) Сортировка результатов

- По % совпадения → сортировка слиянием ( $O(n \log n)$ )
- По времени приготовления → сортировка вставками (если список небольшой).

## Дополнительно

Для данного проекта нужно применить некоторые алгоритмы из этого списка (т.е. просто подобрать подходящие для реализации этого проекта):

- Сортировка методом пузырька
- Сортировка методом простого выбора
- Сортировка методом простых вставок
- Сортировка Шелла
- Сортировка слиянием
- Сортировка Хоара
- Рекурсивные алгоритмы

- Алгоритм прямого поиска подстроки в строке
- Алгоритм Боуера и Мура
- Алгоритм Кнута, Мориса и Пратта
- Алгоритма Крускала
- Алгоритма Прима

## **Технологии и сложность**

- Язык: Python
- Библиотека: json — для хранения рецептов.

## **Дополнительные заметки**

- Нужно изначально занести какое-то кол-во блюд (~20 позиций), чтобы было разнообразие
- Нужен простой интерфейс для данного приложения.