

Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут ім. Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки

### **ЛАБОРАТОРНА РОБОТА № 3\_a**

з дисципліни «Методи оптимізації та планування експерименту» на тему  
«Дослідження генетичного алгоритму»

Виконав:  
студент II курсу ФІОТ  
групи ІО-93  
Бриль Владислав

Перевірів:  
Асистент Регіда П.Г.

# Лабораторна робота 3\_a

**Мета роботи** - ознайомлення з принципами реалізації генетичного алгоритму, вивчення та дослідження особливостей даного алгоритму з використанням засобів моделювання і сучасних програмних оболонок.

## Завдання

Налаштувати генетичний алгоритм для знаходження цілих коренів діофантового рівняння  $ax^1 + bx^2 + cx^3 + dx^4 = y$ . Розробити відповідний мобільний додаток і вивести отримані значення. Провести аналіз витрат часу на розрахунки.

## Теоретичні відомості

Генетичні алгоритми служать, головним чином, для пошуку рішень в багатовимірних просторах пошуку.

- Можна виділити наступні етапи генетичного алгоритму:
- (Початок циклу)
- Розмноження (схрещування)
- Мутація
- Обчислити значення цільової функції для всіх особин
- Формування нового покоління (селекція)
- Якщо виконуються умови зупинки, то (кінець циклу), інакше (початок циклу).

## Виконання роботи

Код програми:

```
package com.lab3a.activity;

import android.os.Build;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
```

```

import com.lab3a.R;
import com.lab3a.execution.EquationSolver;
import com.lab3a.utils.Permissions;

public class MainActivityInflater {

    @RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
    public static void inflate(AppCompatActivity activity) {

        EditText edittext_input_a =
activity.findViewById(R.id.edittext_input_a);
        EditText edittext_input_b =
activity.findViewById(R.id.edittext_input_b);
        EditText edittext_input_c =
activity.findViewById(R.id.edittext_input_c);
        EditText edittext_input_d =
activity.findViewById(R.id.edittext_input_d);

        EditText edittext_input_y =
activity.findViewById(R.id.edittext_input_y);

        Button button_count = activity.findViewById(R.id.button_count);

        View.OnClickListener onButtonCountClick = v -> {

            TextView textview_output_result =
activity.findViewById(R.id.textview_output_result);

            String string_a =
String.valueOf(edittext_input_a.getText());
            String string_b =
String.valueOf(edittext_input_b.getText());
            String string_c =
String.valueOf(edittext_input_c.getText());
            String string_d =
String.valueOf(edittext_input_d.getText());

            String string_y =
String.valueOf(edittext_input_y.getText());

            if (
                string_a.trim().equals("") ||
string_b.trim().equals("") ||
                string_c.trim().equals("") ||
string_d.trim().equals("") ||
                string_y.trim().equals("")
            ) {

textview_output_result.setTextColor(activity.getResources().getColor(R.c
olor.red));

                textview_output_result.setText("Не введені дані");

            } else if (
                string_a.trim().equals("-") ||

```

```

string_b.trim().equals("-") ||
                                string_c.trim().equals("-") ||
string_d.trim().equals("-") ||
                                string_y.trim().equals("-")
    ) {

textView_output_result.setTextColor(activity.getResources().getColor(R.c
olor.red));
        textView_output_result.setText("Неправильно введені
дані");

    } else {

        long a = Long.parseLong(string_a);
        long b = Long.parseLong(string_b);
        long c = Long.parseLong(string_c);
        long d = Long.parseLong(string_d);

        long y = Long.parseLong(string_y);

        EquationSolver solver = new EquationSolver();

        solver.setCoefficients(new long[] {a, b, c, d});
        solver.setY(y);

        solver.solve();

        long[] roots = solver.getRoots();

        Permissions permissions = new Permissions(roots);

        long[][] perms = permissions.getPerms();

        for (long[] perm : perms) {

            if (a * perm[0] + b * perm[1] + c * perm[2] + d *
perm[3] == y)
                roots = perm;

        }

        StringBuilder out = new StringBuilder();

        for (int i = 0; i < roots.length; i++) {
            out.append("X").append(i + 1).append(" =
").append(roots[i]).append("\n");
        }

textView_output_result.setTextColor(activity.getResources().getColor(R.c
olor.green));
        textView_output_result.setText(out);

    }

```

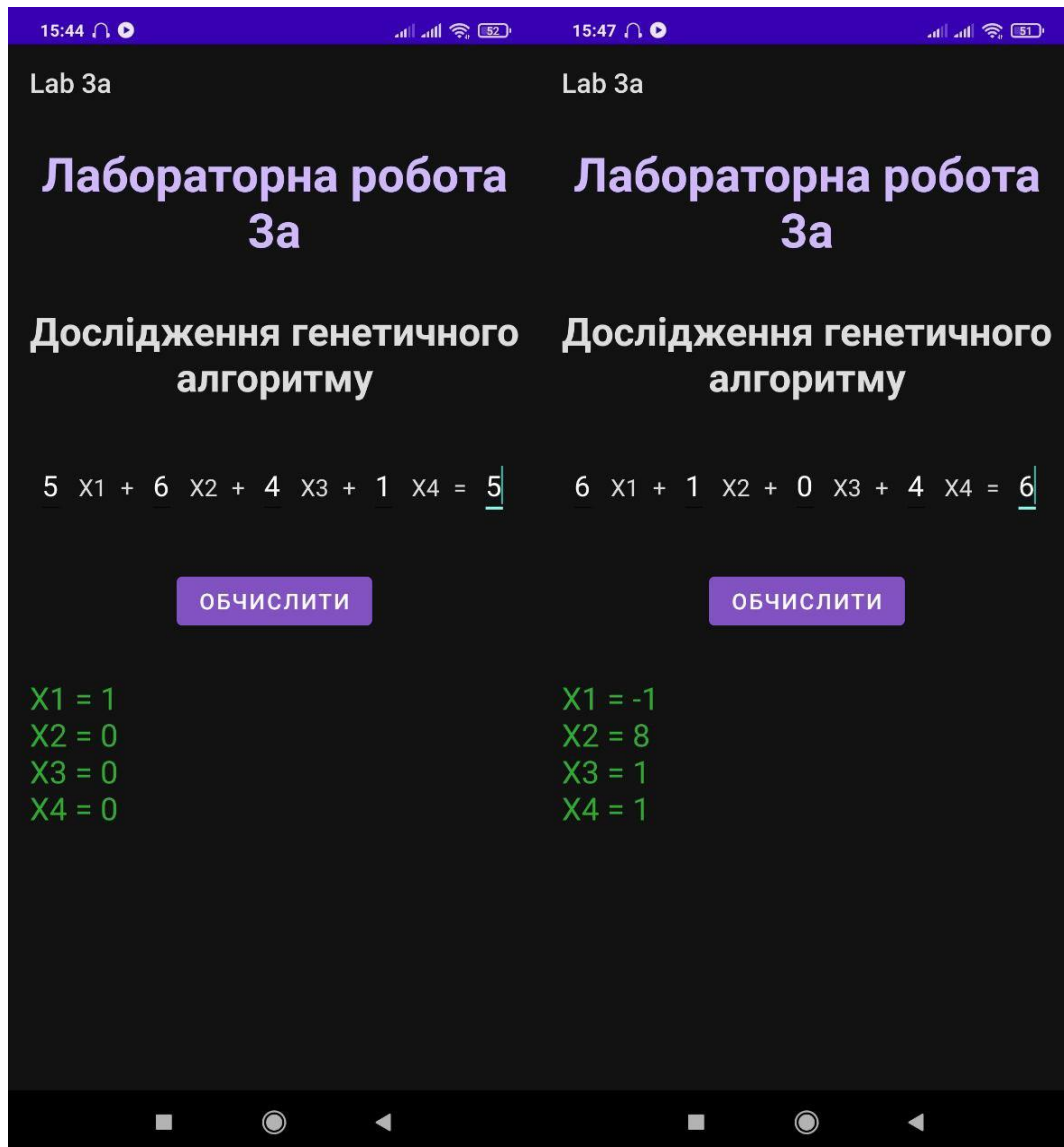
```
};

button_count.setOnClickListener (onButtonCountClick);

}

}
```

## Результати роботи



## Висновок

Виконавши цю роботу, я ознайомився з принципами реалізації генетичного алгоритму, вивчив та дослідив особливості даного алгоритму з використанням засобів моделювання і сучасних програмних оболонок.