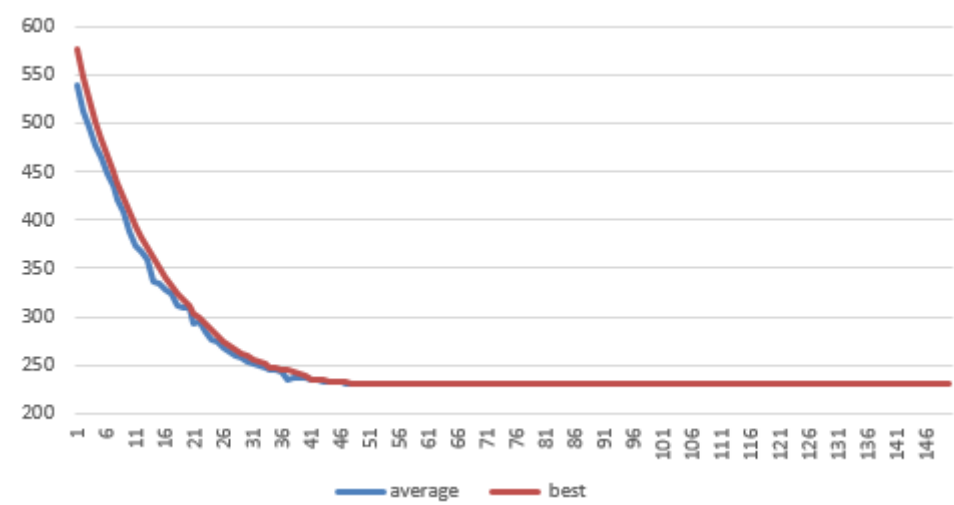
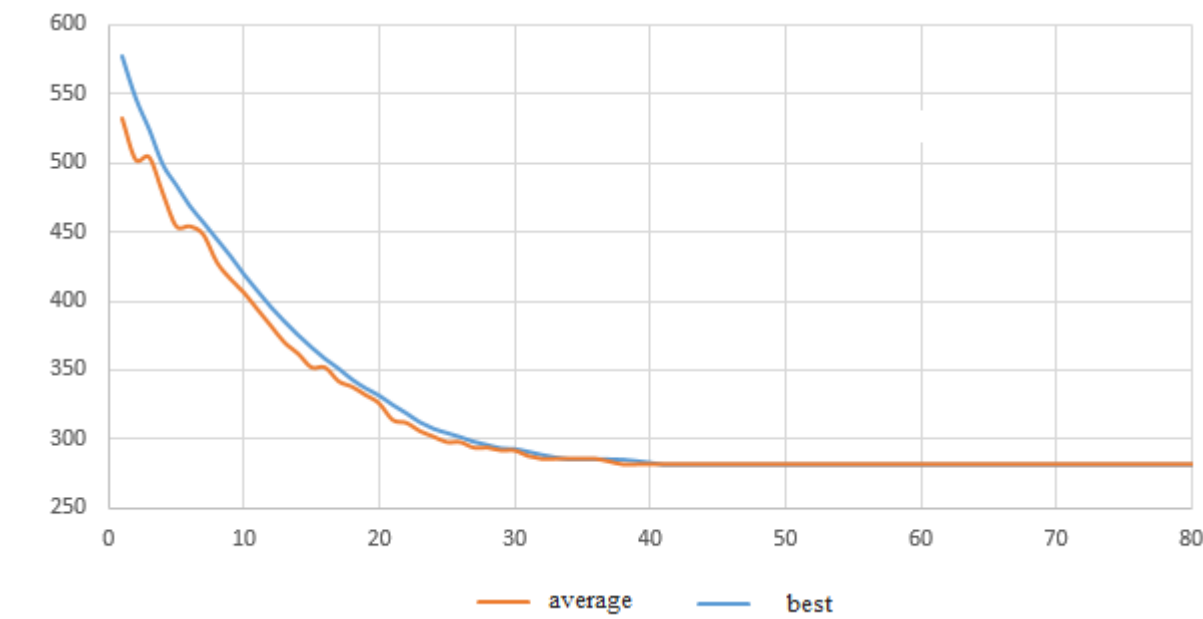


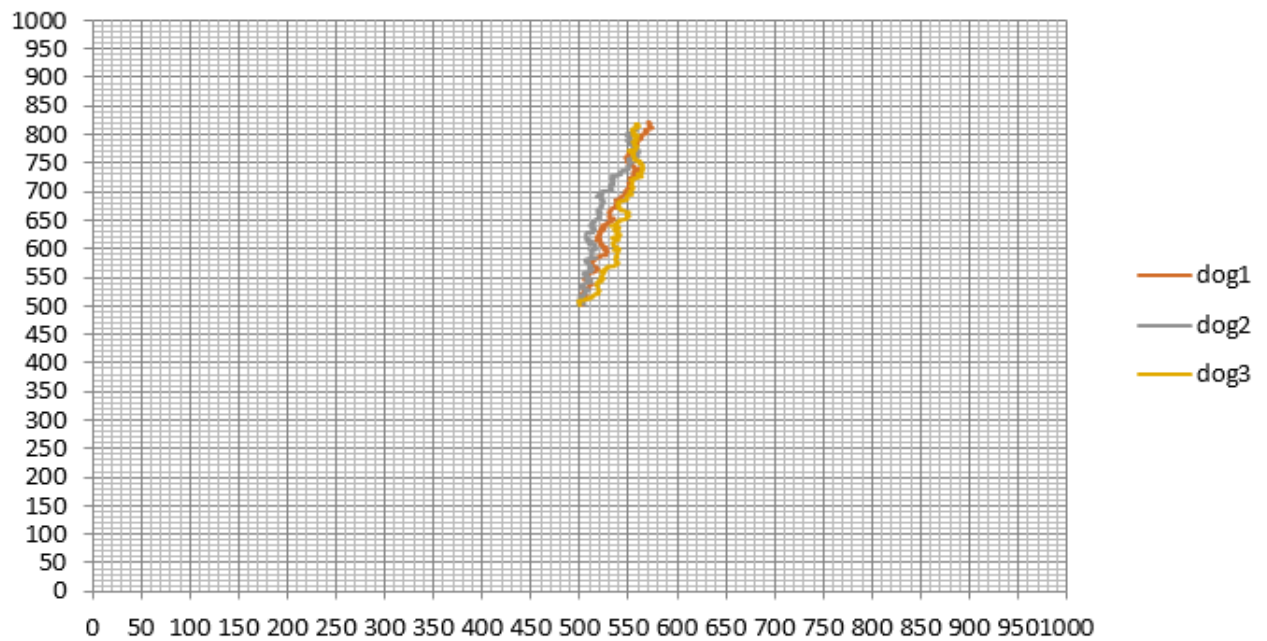
task 2-Kravchuk Maxim

onepoint



Uniform





```
import java.util.Random;
```

```
public class Dog {
```

```
    public static void main(String... args){
        int[][] osob = new int[100][1000];
        Random random = new Random();
        int iter = 0;
        double avr = 0;
        double numb = 0;
        for (int i = 0; i < 100; i++){
            for (int j = 0; j < 1000; j++){
                osob[i][j] = random.nextInt(3);
            }
        }
        for (int i = 0; i < 100; i++){
            for (int j = 0; j < 1000; j++){
                osob[i][j]++;
                avr += osob[i][j];
            }
        }
        System.out.println(avr / (100 * 1000));
        int[] temp = new int[1000];
        int[][] best = new int[50][1001];

        do{
            for (int i = 0; i < 99; i++){
                int min_fitness = i;
                for(int j = i + 1; j < 100; j++) {
```

```

        int fitMIN = stepsToSousage(osob[min_fitness]);
        int fitJ = stepsToSousage(osob[j]);
        if(fitJ<fitMIN)min_fitness = j;
    }

    if (min_fitness != i) {
        for(int k=0;k<1000;k++)
            temp[k]=osob[i][k];

        for(int k=0;k<1000;k++)
            osob[i][k]=osob[min_fitness][k];

        for(int k=0;k<1000;k++)
            osob[min_fitness][k]=temp[k];
    }
}

if(iter==0)
{
    for(int s=0;s<5;s++)
    {
        coord c = new coord();
        c.X= 500;
        c.Y= 500;
        for(int i=0;i<1000;i++) {
            switch (osob[s][i]) {
                case 0:
                    c.Y--;
                    break;
                case 1:
                    c.Y++;
                    break;
                case 2:
                    c.X--;
                    break;
                case 3:
                    c.X++;
                    break;
                default:
                    System.out.println("Error");
                    break;
            }
            System.out.println("X" + s + c.X);
            System.out.println("Y" + s + c.Y + "\n");
        }
    }
}

double AvrFit=0;
int BestFit=0;
int numOfBest=0;
coord c;
for(int i=0;i<50;i++) {

```

```

        c = getCoord(osob[i]);
        if(c.X<0 || c.Y<0)continue;
        numOfBest++;
        for(int j=0;j<1000;j++) {
            best[i][j]=osob[i][j];
        }
        AvrFit += stepsToSousage(best[i]);
    }

    AvrFit = AvrFit/50;
    System.out.println("AvrFit = " + AvrFit);

    BestFit = stepsToSousage(best[0]);

    System.out.println("BestFit = " + BestFit + "\n");

    if(BestFit==0)break;
    int mom=0;
    int dad=0;
    int cat=0;
    int[] frstChild = new int[1000];
    int[] ScndChild = new int[1000];
    for(int i=0;i<100;i++) {
        do
        {
            mom = random.nextInt(numOfBest);
            dad = random.nextInt(numOfBest);
        }while(mom == dad);
        cat=random.nextInt(1000);
        for(int k=0;k<cat;k++) {
            frstChild[k]=best[mom][k];
            ScndChild[k]=best[dad][k];
        }

        for(int k=cat;k<1000;k++)
        {
            frstChild[k]=best[dad][k];
            ScndChild[k]=best[mom][k];
        }
        int fitness1=0;
        int fitness2=0;

        fitness1 = stepsToSousage(frstChild);
        fitness2 = stepsToSousage(ScndChild);

        if(fitness1<fitness2)
        {
            for(int k=0;k<1000;k++)
                osob[i][k]=frstChild[k];
        }
        else
        {
            for(int k=0;k<1000;k++)
                osob[i][k]=ScndChild[k];
        }
    }

```

```

        }

        }
        iter++;
    }while (iter < 100);
}

static class coord{
    int X;
    int Y;
}

public static int stepsToSousage(int[] dog){
    coord c = new coord();
    c.X = 500;
    c.Y = 500;
    for(int i=0;i<1000;i++) {
        switch (dog[i]) {
            case 0:
                c.Y--;
                break;
            case 1:
                c.Y++;
                break;
            case 2:
                c.X--;
                break;
            case 3:
                c.X++;
                break;
            default:
                System.out.println("Error");
                break;
        }
    }

    int steps=0;
    steps=Math.abs(c.X - 800) + Math.abs(c.Y - 800);
    return steps;
}

public static coord getCoord(int[] dog){
    coord c = new coord();
    c.X= 500;
    c.Y= 500;
    for(int i=0;i<1000;i++) {
        switch (dog[i]) {
            case 0:
                c.Y--;
                break;
            case 1:
                c.Y++;
                break;
            case 2:

```

```

        c.X--;
        break;
    case 3:
        c.X++;
        break;
    default:
        System.out.println("Error");
        break;
    }
}
return c;
}

public static int isFitnessGood(int[][] osob){
    int count = 0;
    for (int i = 0; i < 50; i++){
        count = 0;
        for (int j = 0; j < 1000; j++){
            count += osob[i][j];
        }
        if(count == 1000){
            return i;
        }
    }
    return -1;
}
}

```