

Programm (Write in Java)

My program work while values no more 30 repeats.

Main.java:

```
package com.company;
import java.util.*;
import static java.util.Collections.swap;
public class Main {
    public static Integer[][] generation = new Integer[100][1000];
    public static Integer[][] good = new Integer[50][1000];
    public static Integer[][] childe = new Integer[50][1000];
    public static Integer childCount = 0;
    public static Integer checkCount = 0;
    public static Integer prevMax = 0;
    public static void main(String[] args) {
        createGenerate();
        while(true) {
            Arrays.sort(generation, new Comparator<Integer[]>() {
                @Override
                public int compare(Integer[] o1, Integer[] o2) {
                    Integer int1 = rowCount(o1);
                    Integer int2 = rowCount(o2);
                    return int2 - int1;
                }
            });
            Integer max = rowCount(generation[0]);
            if(prevMax.equals(max))
                checkCount ++;
            else
                checkCount = 0;
            prevMax = max;
            System.out.print(max + " ");
            System.out.println(average());
            if(checkCount > 30) {
                return;
            }
            setGood();
            Random random = new Random();
            childCount = 0;
            for (int i = 0; i < 25; i++) {
                createChilde(good[random.nextInt(50)], good[random.nextInt(50)]);
            }
            createNewGeneration();
        }
    }
}
```

Function for generate start generation:

```
public static void createGenerate() {
    Random random = new Random();
    for(int i=0;i<100;i++) {
        for(int j=0;j<1000;j++) {
            if(random.nextDouble() >=0.5)
                generation[i][j] = 1;
            else
```

```

        generation[i][j] = 0;
    }
}

```

Function for count "1" in one chromosome

```

public static Integer rowCount(Integer[] obj) {
    Integer count = 0;
    for(int i=0;i<1000;i++) {
        count += obj[i];
    }
    return count;
}

```

Function for creating good amiboos 'from' all

```

public static void setGood() {
    for(int i =0; i< 50; i++) {
        good[i] = generation[i];
    }
}

```

Function for cross 'parents' and create 'childs':

```

public static void createChildes(Integer[] parent1, Integer[] parent2) {
    Random random = new Random();
    Integer delimiter = random.nextInt(1000);
    Integer[] child1 = new Integer[1000];
    Integer[] child2 = new Integer[1000];
    for(int i = 0;i<1000;i++) {
        if(i<=delimiter)
            child1[i] = parent1[i];
        else
            child1[i] = parent2[i];
    }
    for(int i = 0;i<1000;i++) {
        if(i<=delimiter)
            child2[i] = parent2[i];
        else
            child2[i] = parent1[i];
    }
    childes[childCount] = child1;
    childCount++;
    childes[childCount] = child2;
    childCount++;
}

```

Function for create new generation from good and from childs (good+childs)

```

public static void createNewGeneration() {
    for (int i=0;i<100;i++) {
        if(i<50)
            generation[i] = good[i];
        else
            generation[i] = childes[i-50];
    }
}

```

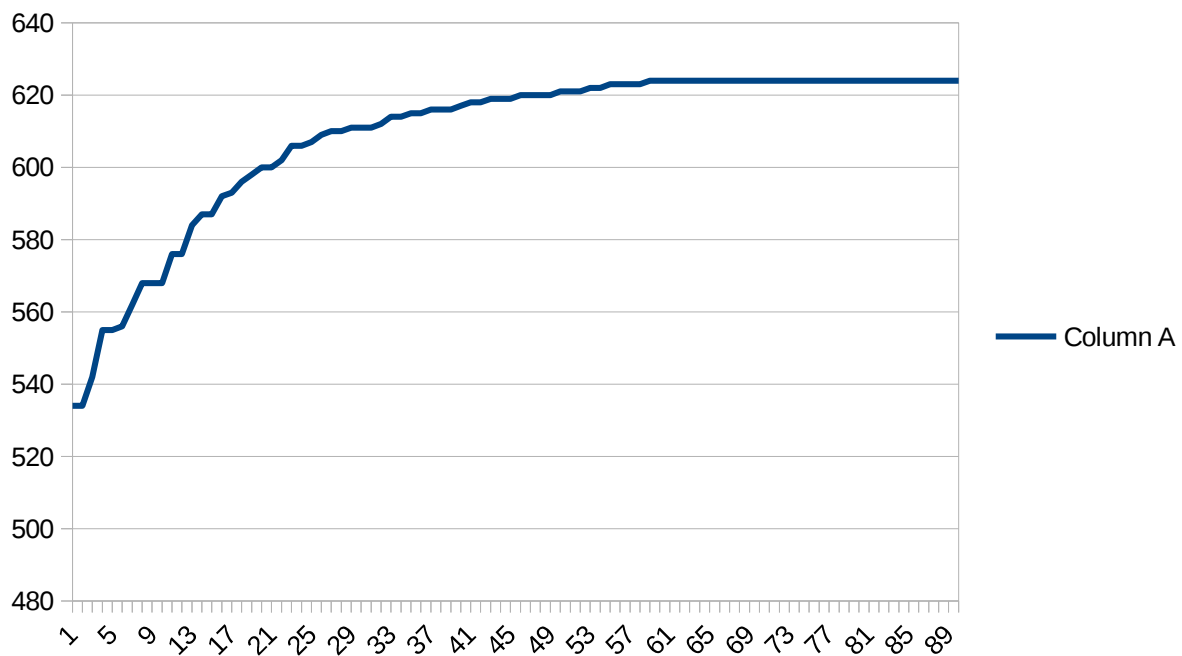
```
}
```

Function for count average sum of '1' in all generation

```
public static Integer average() {  
    Integer s = 0;  
    for (int i=0;i<100;i++) {  
        s+= rowCount(generation[i]);  
    }  
    Integer average = s / 100;  
    return average;  
}  
}
```

After completing program and analyze result, we can see, that algorithm create chromosomes with 62.4% good gens. Average value equals with maximum value, but have less steps by iterations. When we run program again, we can get another results, its depends on random function on JVM.

Graph for greatest chromosomes



Graph for greatest average values from generation

