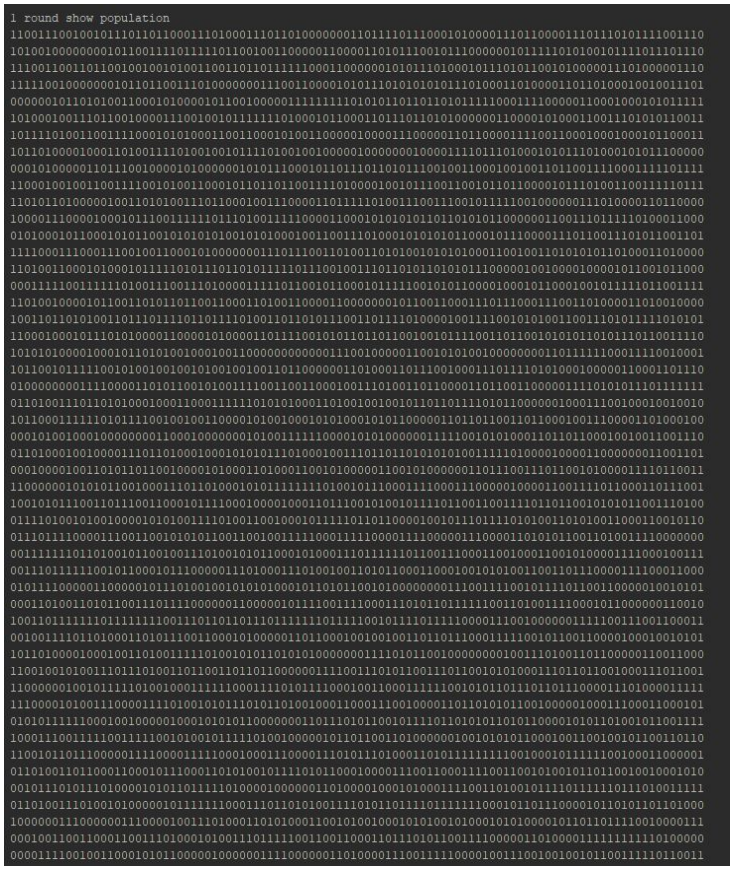
หลักการ GeneticAlgorithm

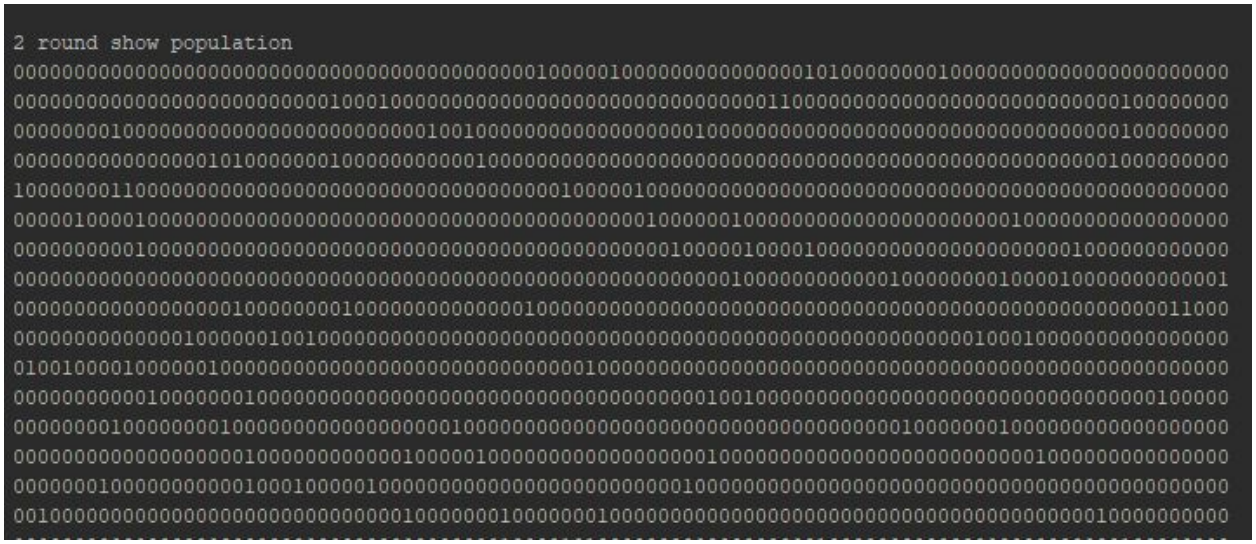
1. เราจะรับ ค่าจาก
Matrix จาก RESULT เข้ามา

2.เรา ดูค่า ที่เข้ามาคือ
ขนาด matrix เป็น 100 จำนวนรอบที่ทำคือ 200 รอบ แล้ว จำนวนที่เป็น สถานีคือ 5

3.ในรอบแรกของการสุ่ม เราจะสุ่ม 0 กับ 1 โอกาสเป็น 50/50



4.ในรอบต่อมาเราจะสุ่ม ให้เป็น 1 ตามจำนวนสถานี
ในตัวอย่างนี้คือ 5
โดยในการสุ่มเราจะสุ่มจากจุดที่เป็น 1 อยู่แล้ว



หลังจากนั้นเราจะ set fitness จากจุดที่เป็น 0 ไปหาจุดที่เป็น 1 ทุกตัว หนึ่งจุดที่ใกล้ที่สุด มาบวกใน fitness

ค่าที่ได้

```
7896
8671
7234
8977
7439
8333
8759
8407
11195
Total fitness 839178
Mean fitness 8391.78
```

หลังจากนั้นมันจะทำการเข้า
Selection, Crossover และ Mutation
จะเปลี่ยนจากจุดที่มีเยอะออกไป

Selection

```
// Tournament Selection
for (int j=0; j < popsize; j++) {
    int champion = rand.nextInt(popsize);
    int contender = rand.nextInt(popsize);
    if (population[champion].fitness>population[contender].fitness) //>
         champion = contender;
    for (int k=0; k < lchrom; k++)
         offsprings[j].gene[k]=population[champion].gene[k];
}
```

Crossover

```
// One-Point Crossover
for (int i = 0; i < popsize; i = i + 2) {
   if (rand.nextDouble() < crossoverRate) {
        int splitPoint = rand.nextInt(lchrom);
        for (int j = splitPoint; j < lchrom; j++) {
            int temp = offsprings[i].gene[j];
            offsprings[i].gene[j] = offsprings[i + 1].gene[j];
            offsprings[i + 1].gene[j] = temp;
        }
   }
}
```

Mutation

```
// One-Point Mutation
for (int i = 0; i < popsize; i++) {
   if ( rand.nextDouble() < mutationRate) {
       int mutationPoint = rand.nextInt(lchrom);
       if(offsprings[i].gene[mutationPoint] == 1)
          offsprings[i].gene[mutationPoint] = 0;
       else
          offsprings[i].gene[mutationPoint] = 1;
   }
}
```

ต่อมา ก็จะเก็บค่าที่ได้เอาไว้ ใช้ในรอบต่อ ไป

```
// Copy offspring to the next generation
for (int i = 0; i < popsize; i++) {
   for( int j = 0; j < lchrom; j++) {
          population[i].gene[j]=offsprings[i].gene[j];
   }
}
```

หลังจากนั้นเราก็จะทำซ้อมค่า ที่มี จำนวน 1 มากกว่า และ น้อยกว่าค่าสถานี
การสุ่มจากจุดที่เป็น 1 อยู่เดิม ให้ได้ ตามจำนวนสถานี

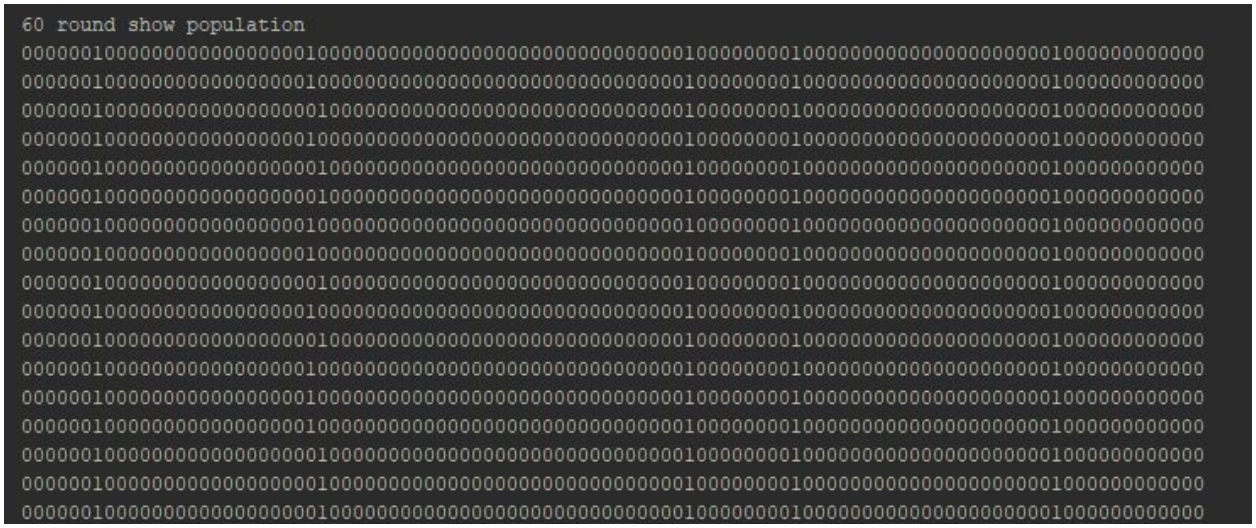
```
int amount_choose = P;
for (int i = 0; i < popsize; i++) {
  ArrayList<Integer> indexOfBase = new ArrayList<>();
  indexOfBase.clear();
 for (int j = 0; j < lchrom; j++) {

    if(population[i].gene[j]== 1){
          indexOfBase.add(j);
         population[i].gene[j]= 0;
    }
   }
 Collections.shuffle(indexOfBase);

 for(int p=0;p<amount_choose;++p) {
    // System.out.println(indexOfBase.get(p));
    if(p>=indexOfBase.size()){
        int randomp = rand.nextInt(lchrom);
        for(;;){
            if(indexOfBase.contains(randomp)){
                randomp = rand.nextInt(lchrom);

            }
            else{
                break;
            }
        }
        population[i].gene[randomp]=1;
        continue;
    }
     population[i].gene[indexOfBase.get(p)] =1;
 }

}
//if(gcounter==maxgen-1){
```

3. ในรอบต่อต่อไปเราจะทำแบบนี้ไปเรื่อยเรื่อย
จุด ที่จะค่อยค่อยเข้า หากัน แล้วค่าที่ได้ก็จะค่อยค่อยน้อยลง

```
20 round show population
00100010000000000000000000000000000000000000000000000000100000000100000000000100000000000000000000
00000010000000000000000010000000000000000000000000000000100000000100000000000100000000000000000000
00000010000000000000000000010000000100000000000000000001000000000000000000001000000000000000000000
00100010000000000000000000010000000000000000000000000001000000000100000000000000000000000000000000
00000010000000000000000010000000000000000000000000000001000000000100000000000100000000000000000000
00010010000000000000000000000000000000000000000000000001000000000000000000001000000000000000000010
00100010000000000000000000000000000000000000000000000001000000000100000000000100000000000000000000
00000010000000000000000010000000000000000000000000000001000000000100000000000100000000000000000000
00000010000000000000000010000000000000000000000000000001000000000100000000000100000000000000000000
00000010000000000000000010000000000000000000000000000001000000000100000000000100000000000000000000
00100010000000000000000000000000000000000000000000000001000000000100000000000100000000000000000000
00000010000000000000000010000000000000000000000000000001000000000100000000000100000000000000000000
00000010000000000000000010000000000000000000000000000001000000000100000000000100000000000000000000
00000010000000000000000010000000000000000000000000000001000000000100000000000100000000000000000000
00000010000000000000000010000000000000000000000000000001000000000100000000000100000000000000000000
00000010000000000000000000000100000000000000000000000001000000000100000000000100000000000000000000
00000010000000000000000010000000000000000000000000000001000000000100000000000100000000000000000000
00000010000000000000000010000000000000000000000000000001000000000100000000000100000000000000000000
00000010000000000000000010000000000000000000000000000001000000000100000000000100000000000000000000
00100010000000000000000000010000000000000000000000000001000000000000000000001000000000000000000000
00000010000000000000000010000000000000000000000000000001000000000100000000000100000000000000000000
00000010000000000000000010000000000000000000000000000001000000000100000000000100000000000000000000
```

```
6134
6218
6519
6134
Total fitness 625409
Mean fitness 6254.09
```

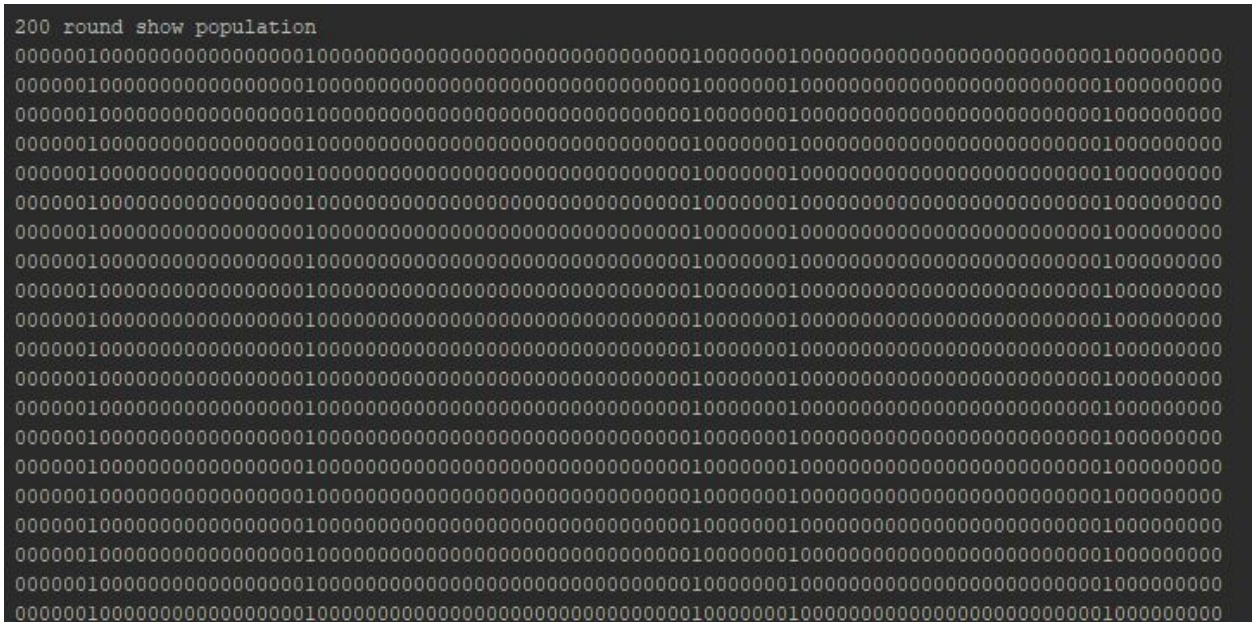
```
60 round show population
```



```
6134
6218
6519
6134
Total fitness 625409
Mean fitness 6254.09
```

4.หลังจากครบ 200 รอบ

```
200 round show population
```



ค่าที่ได้

```
5876
5876
5876
5876
5876
5876
5876
Total fitness 587600
Mean fitness 5876.0
```

จุดที่เป็นสถานี

```
station is 7
station is 25
station is 57
station is 65
station is 91
```

ปล. ในการ run แต่ละครั้ง ค่าที่ได้จะ คาดเคลื่อน บวกลบ ไม่เกิน 100;
เสริม ไม่ต้องใส่ในรายงานกับ slide //มันเคลื่อนจากการที่เราสุ่มแต่ละรอบ จากจุดที่เป็น 1 อยู่แล้ว