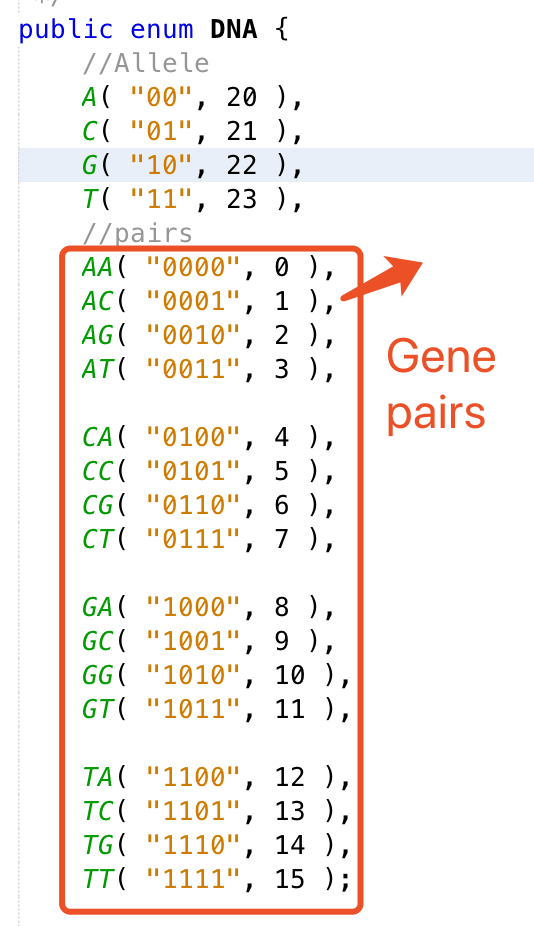
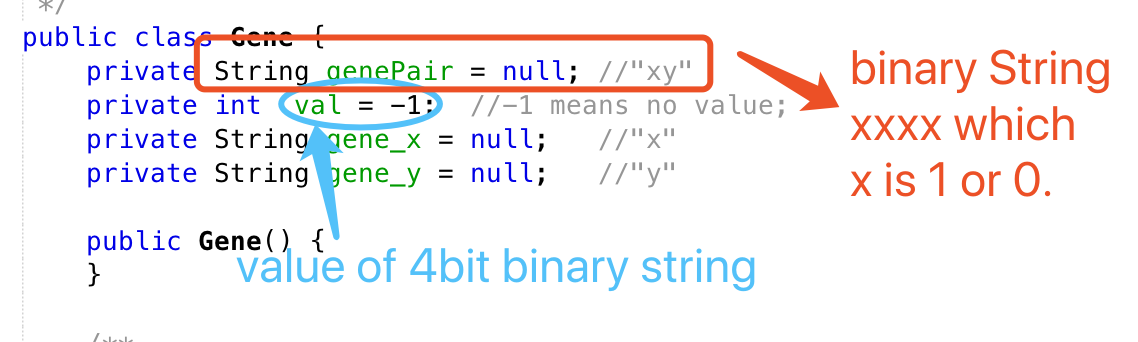
INFO6205 Final Project\_No.09 Bo Han (001815357) & Junyuan Gu (001825583)

Problem: Travelling Sales Problem, find a shortest cycle between city nodes.

1. We use “ACGT” to implement a genetic code, where ‘A’,’C’,’G’,’T’ are alleles and there are 16 kinds of gene combination(thus use 4 bits binary String as key). The genotype is 16 bit width, which stands 4 gene pairs. Because we use 16 bit genotype to stand for city code( assuming 100 cities, the city code is from 0 to 99), the most 8 significant bits are always zero.



2. Gene expression: We use 4-bit binary String as well as integer value to stand for Gene, therefore 4 Genes ( 16bit binary String )stands for the code of a city node(from 0 to 99).



3. Fitness function: we use Position(x, y) thus compute the distance between two cities. Of course the fitness is the inverse of the distance. And “coor.txt” is the initialization data of city positions.

4. About sort function: we provide quicksort and mergesort. Actually we use mergesort() method.

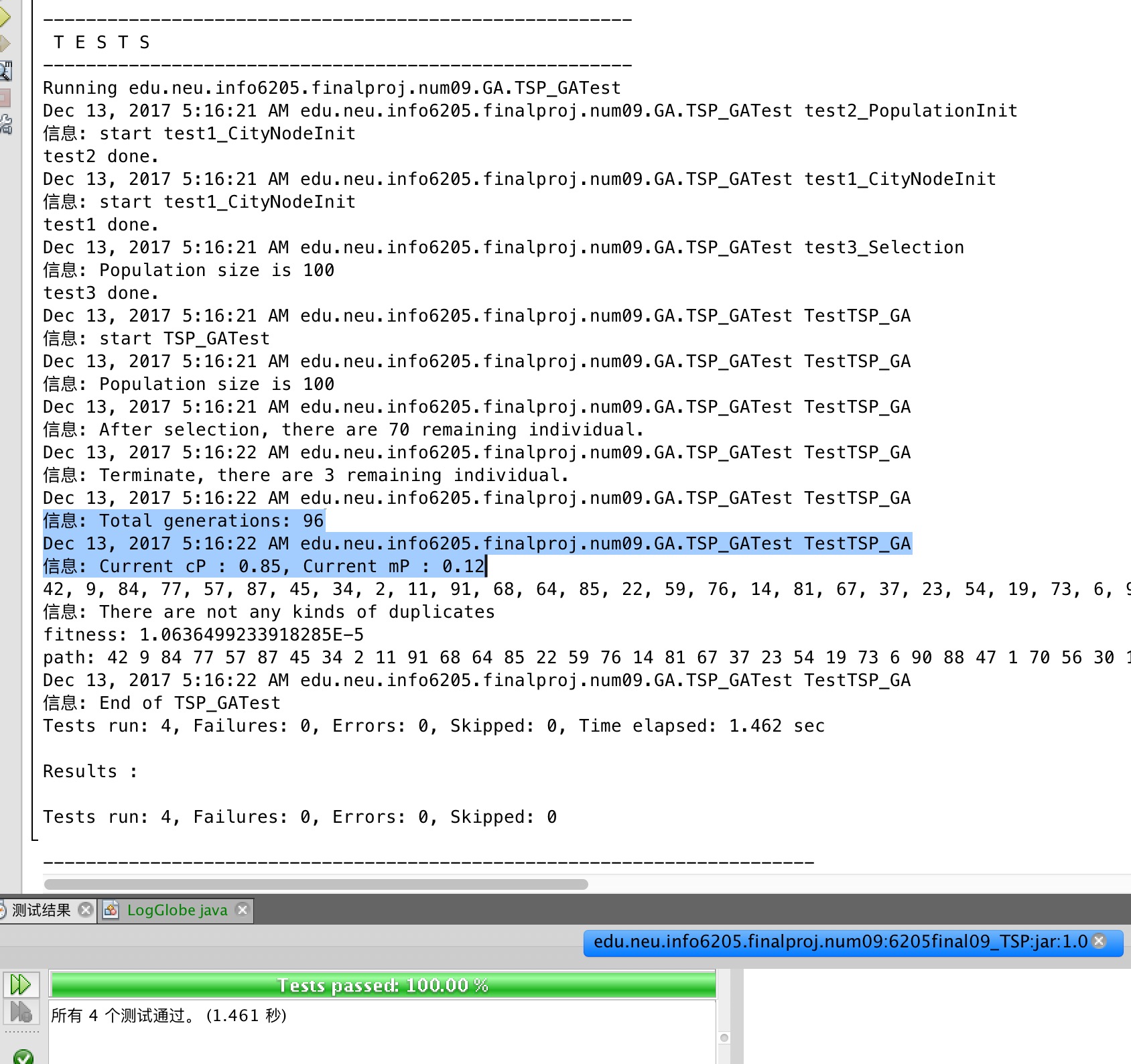
About log: we use java.util.logging.Logger to log the unit test.

About unit test: we test initialization of city and Population, selection of Population respectively and the whole evolution.

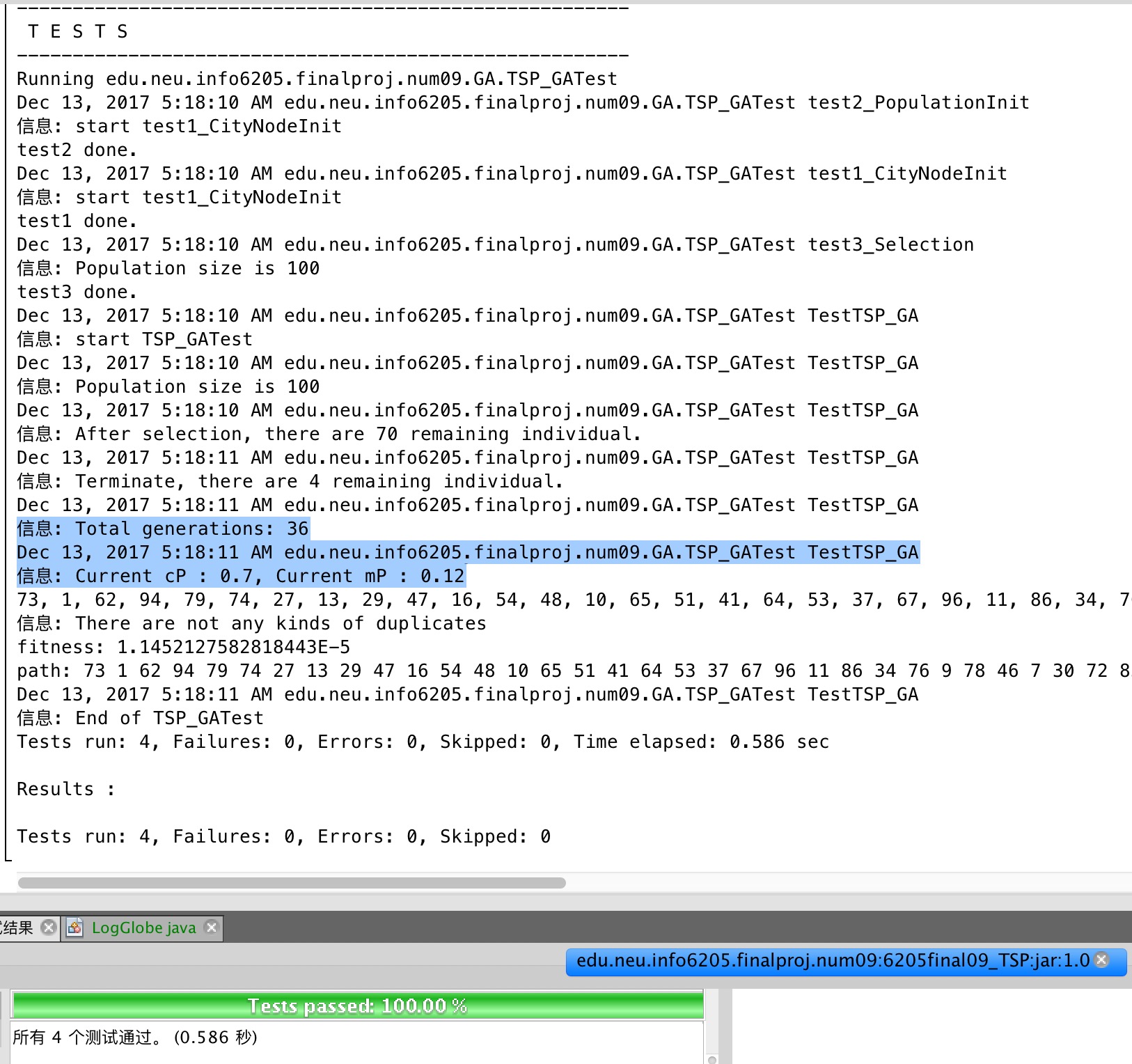
5. Findings: ( our city nodes are 100, and initial population size is 100, and we cull the 30% worst case using sort by fitness every generation.)

a. when the cP(crossover Probability) is less than 0.75, the whole population size will converge quickly(the iteration is about 30 to 38 times). In this case we only get quite a small number of possible good solutions. The solution space is huge since there are 100 city nodes.

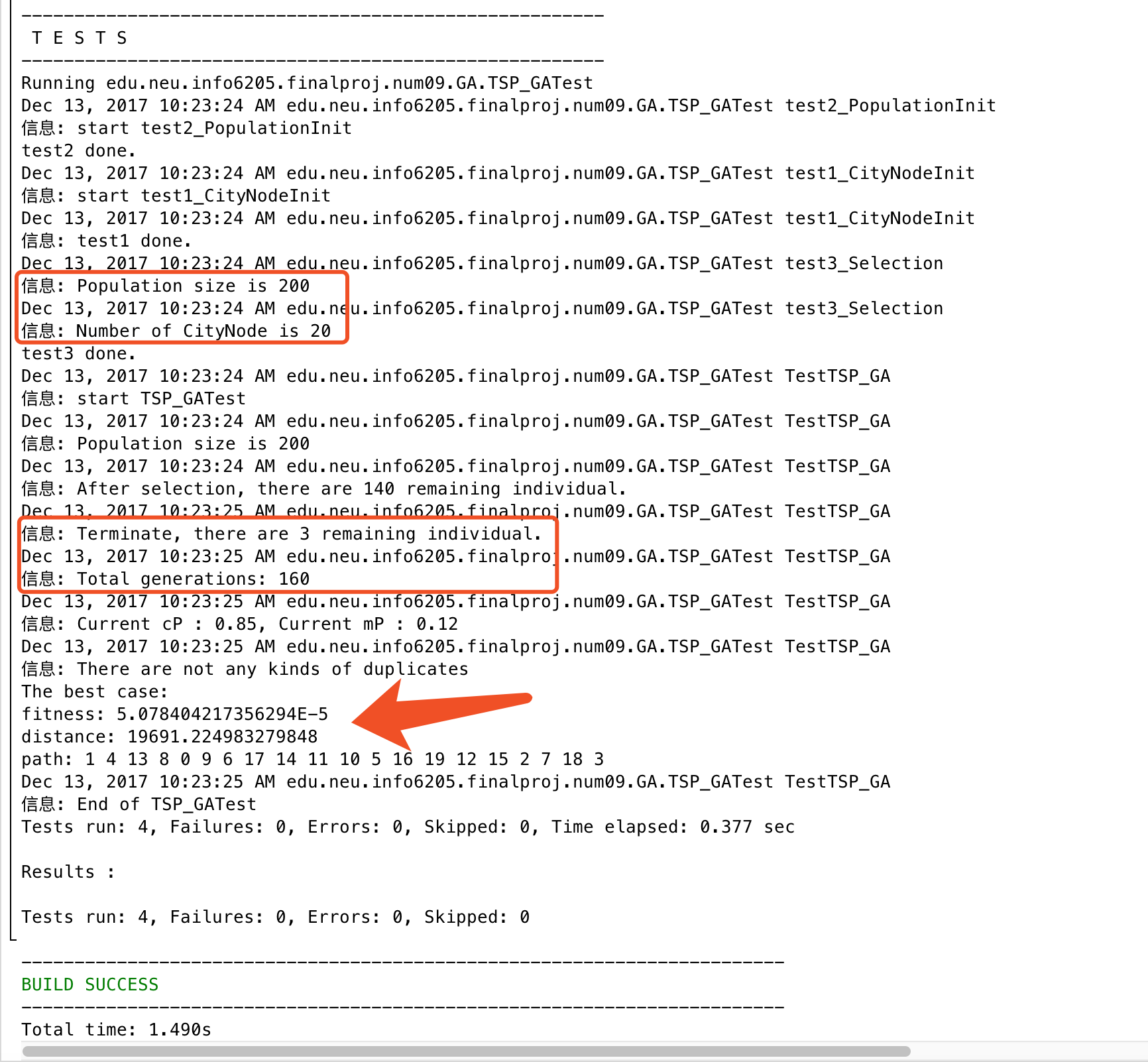
Current cP: 0.85, mP:0.12



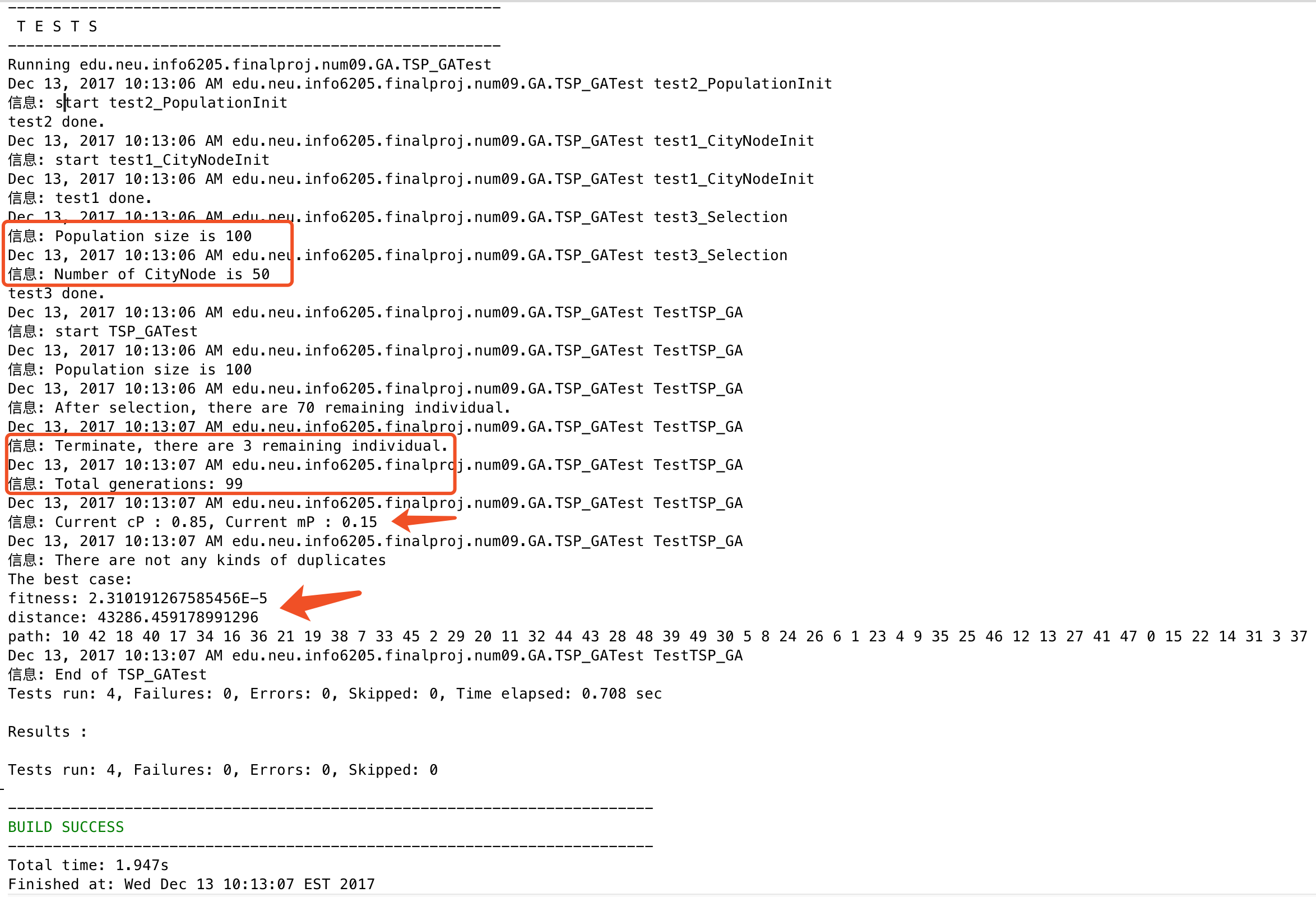
Current cP: 0.7, mP:0.12



Best case: When number of city node is 20, P(crossover) = 0.85, P(mutation) = 0.12.

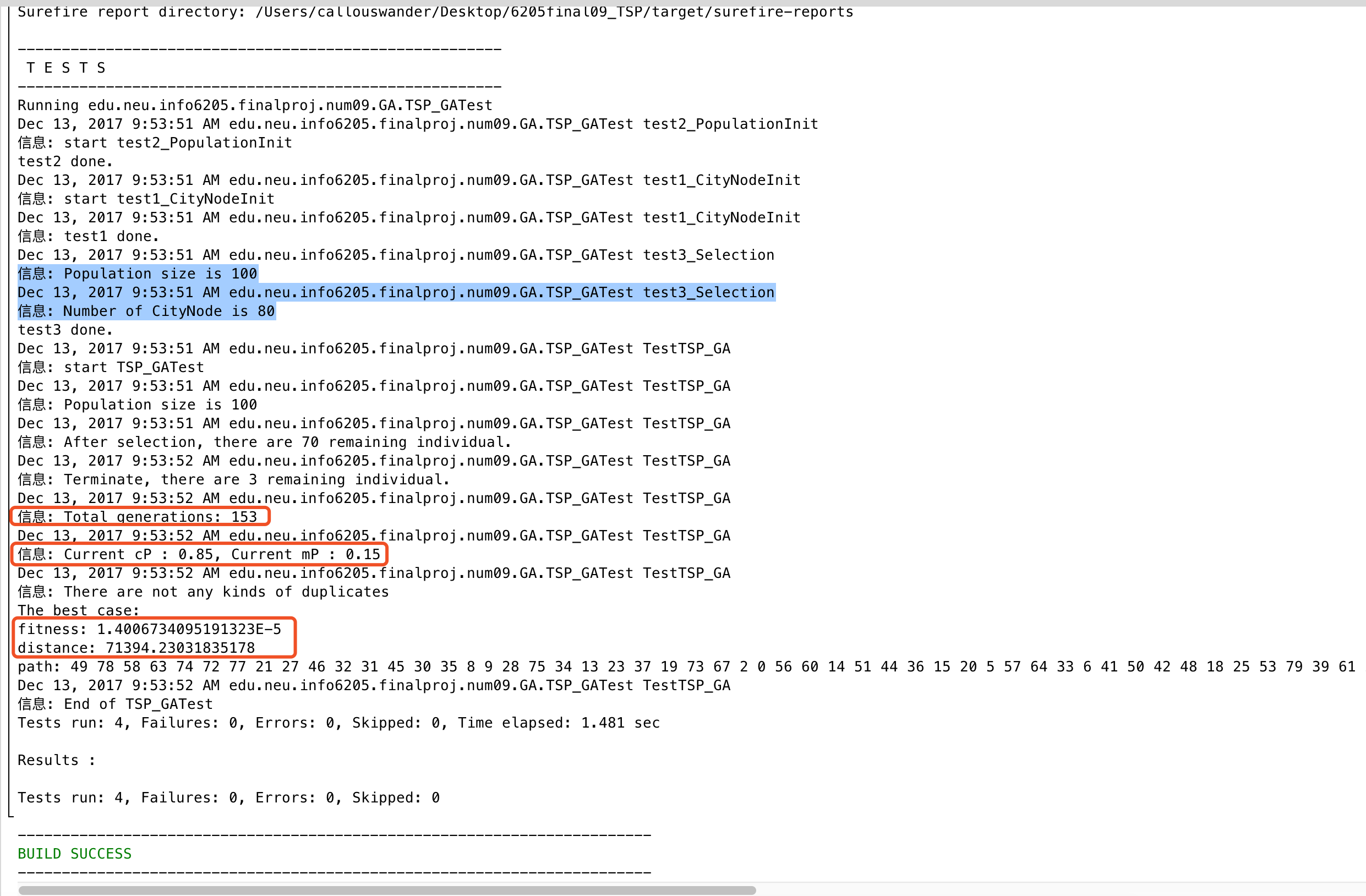


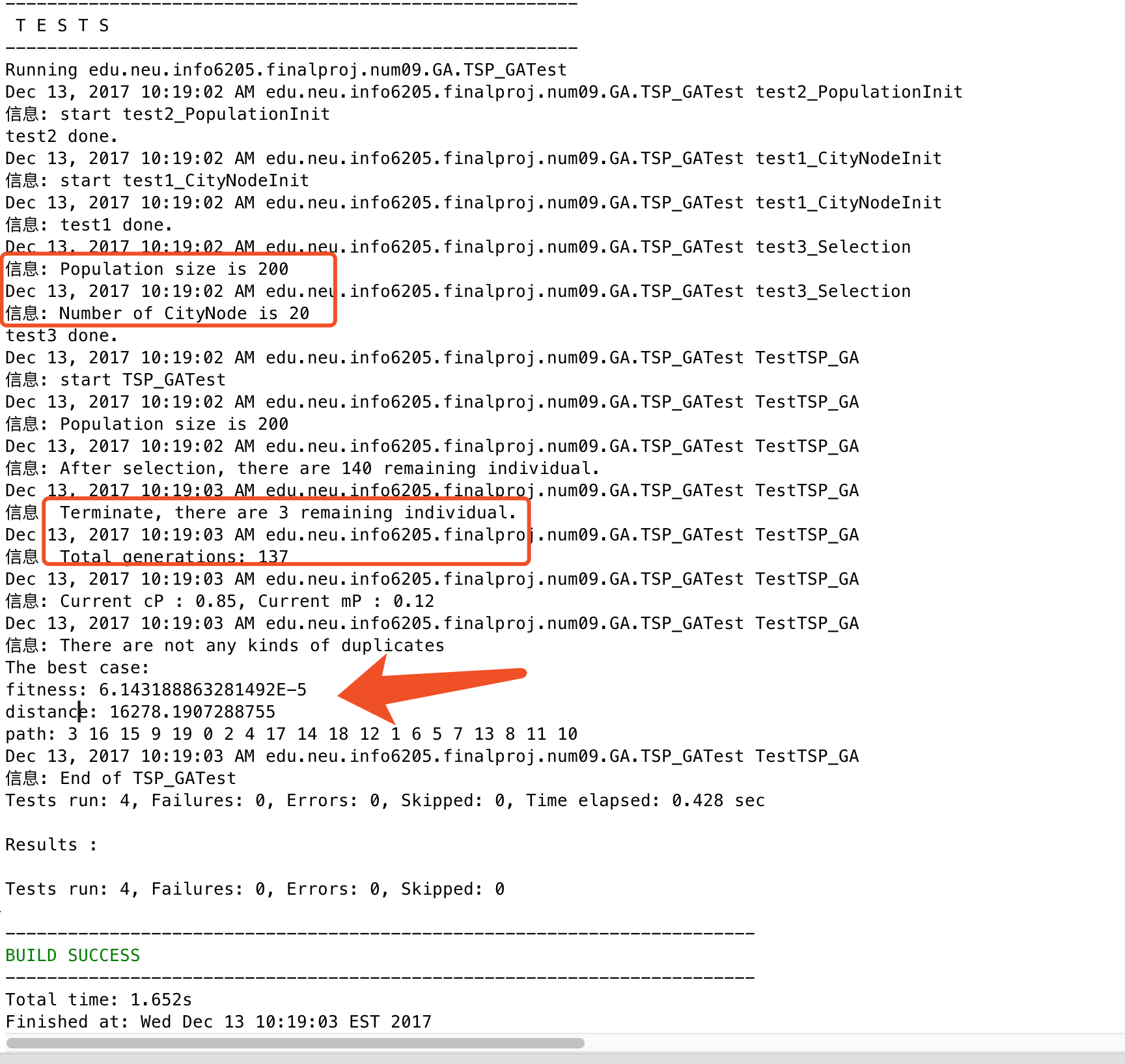
b. When the cP is about 0.85, the convergency speed will be slow(the iteration is about 100 to 130).



c. When we reduced the city node, like from 100 to 80, to 50, and to 20, the iteration will become large and it covers more scope of possible good solutions.

As follows, city node = 80 and the iteration is 150





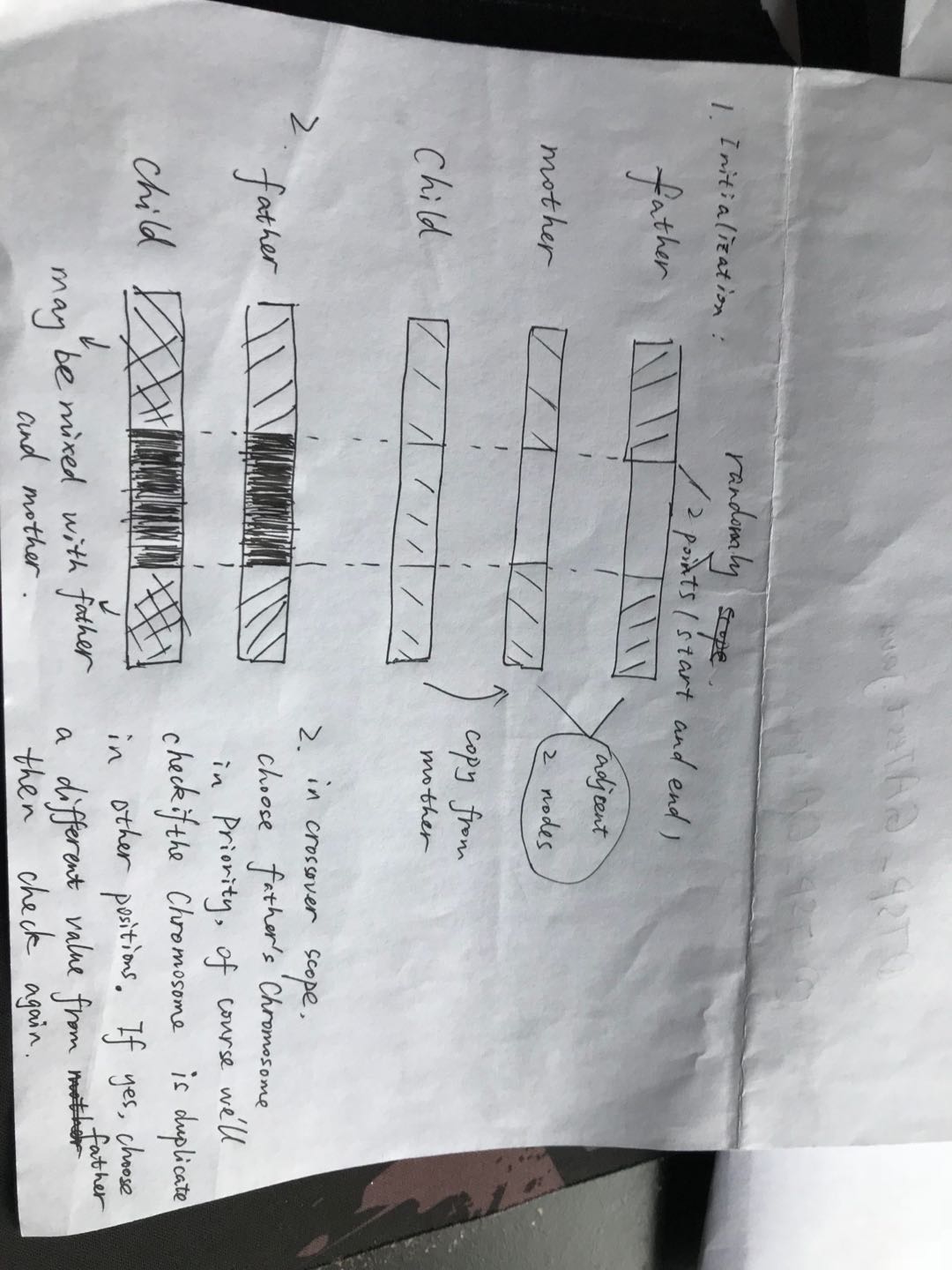
6. About Crossover solution: Firstly, we stuck at this method, we could not perform a precise way to identify the duplicate CityNodes. Finally, we find solution as follows:

A. Two individuals send to crossover(), method, use mother to initialize this child.

B. Generate two random indexes to indicate the start and end point

C. Copy the Chromosomes between start point and end point from father.

D. We implement additional operations, actually the extra ArrayList and contains() method to check the duplicate chromosomes.



Hope to explain it clearly, but I feel a little bit difficult to express literally.

7. Conclusion

7.1 The probability of crossover should be large enough( ie. >=0.8 via experiments and observations ). If not, the population size will converge quickly and iteration won’t be enough to cover the solution space.

7.2 The population size should be big while the number of city node should be small(say 200 population size and 20 city nodes). In this case, the genetic algorithm could cover more scope of good/better solutions.

7.3 We did stuck at the crossover where we didn’t deal with duplicate elements correctly at first. Fortunately we solved it through ArrayList and contains() method but it took us quite a lot of time in debugging whether crossover() is correct simply because we override the compareTo() but forgot to override equals() method in Chromosome.java, which seemed a little bit ridiculous.

8.Further improvement

8.1 We didn’t designate the initial node of city while TSP requires that salesman should start from a designated city and return back to this city. What we implemented is that find the shortest cycle during N city nodes.

8.2 When we chose the better fitness case of population, we used quick sort/ merge sort. But it seemed not efficient when population is just not so large, and more importantly, if the population size is large(eg. 1000 or 5000), using sort first and then find the fitness ones is not so good. We could use priority queue to pick up a few better solutions.

8.3 We wasted more than 8 MSB(most significant bits )of Chromosome(total is 16) cause the gene expression didn’t require much.

Finally thank you again, Professor.

Have a good holiday!