Program1

Ching-Wei Lin

#987563037

Debugger:

This program is a really huge one. I built it up in two separate part and then I combined them together. The first part is mapping and the second one is package list. For the gdb performance in the first part, I used it to check the error of giving the wrong address in adjacent list. The most common error is giving an address which is too big. For the performance of g++ compiler, out of range is a type of segmentation fault. Also, reaching an address which has not been allocated is a common issue. In the second part of the program, I faced some over traversal problem which cause me access the data which is not allocated yet, but compared with the first part, there is less segmentation fault in the second part. I guess the reason why is that there is not much list to traverse. The mapping section did the traversal movement for it, so after I implemented the mapping part, I just have to call the function correctly. Moreover, since g++ compiler will check if my function calls and the argument in it are correct, I would know where is the error and how could I fix it.

Analyzing:

As I mentioned above, the first part of the program is mapping. In the mapping section, I created an array of pointer to cities and also each city has its own adjacent list with the detail of that path. Then, in the beginning of the program, I ask the user to enter all the information for the map and there are two factors, city and connection. City is built as an object which contains its name and its connection list. Each city has its own connection list which is built as a linear link list, so one city can have multiple connections. Connection contains some information, such as type of road, speed limit, length and status. After creating all the cities, I start to build up the connections between each city, which can avoid connect to a null city causing segmentation fault. The second part is package list. I created a basic class as delivery class and there are three classes, express, standard, drone, inherent it. Then, I created three kind of lists for each of them and ask the user to add the package he wants to delivery. They will be assigned to the corresponding list with their information such as name, weight and destination. After all the packages are all set, the program will display the three list with all the information and the calculation result which express has estimate delivery fee and standard has estimate delivery time. In the calculation function, it’s the part I can do it better if I have more time. The perfect searching algorithm is so complicate that I just did a simple one. I assume all the city are all one a straight line, so if you want to go from a to b, you just have to consider the path between those two. For example, if you want to go from 1 to 5, the shortest path is 1-2-3-4-5. Then, the length of the path is 1-2 + 2-3 + 3-4 + 4-5. For the object oriented design, I used it on the delivery class and the three derived. It is a very convenient design that I don’t have to implement three same functions in each of the class. I just have to implement the basic class and then every derived object could use it. However, building up the constructor is complicated. I have to consider about the initialized list, copy constructor, the constructor with argument. But, overall the program help me learn a lot!