After having the user enter the file name, the program then reads in all the lines in the input file as a list of tuples.

The input list is then sorted by lowest value of either of the entries of the tuple. Therefore, the tuple (9, 1) would come before (2, 3) since 1 is smaller than either 2 or 3.

Next, we have a list called tables that contains all the people that can sit with each other. The tables list is a list of sets. Each set contains the people that are friends and can sit with each other. For each tuple in the input list, we see if either person is already at a table. If so, then the other person is added to the same table.

Otherwise, we create a new table and seat both people there. Since the input list is sorted, then it is not possible that two sets will need to be combined later on as the input list is traversed, for the smaller number in the relationship will already be seated at the correct table.

Sorting the input list takes O(nlogn) time. Since sets are being used, checking if a person is at a specific table takes O(1) time. However, since there can be at most g sets in the tables list, then it takes O(g) time to see if someone is already assigned a table. Since we have to iterate through the input list which has n elements, then assigning people to a table will take O(n \* g). Therefore, this solution runs in O(nlogn) + O(n\*g) time. If g is small in comparison to n, then this can be simplified to O(nlogn).