

### Algorithm Description

I would look to improve equity by trying to place blocks of ticket purchases in the same row instead of randomly placing the regular economy seats after the economy plus have been seated.

I would first create a method which would break down the remaining empty seats into a new dictionary (***remaining\_seats***). I would do this by stripping away the seats that are already filled and then evaluating each row by the number of remaining available seats. The key:value pairs of the dictionary would be ***row:number of available seats***. (example - row 1: 2, row 2: 3, row 3: 1). Then I would create a method to cross reference the ***economy\_sold*** dictionary values against the ***remaining\_seats*** dictionary values to look for matches. When a match is found, that block of seats would be assigned to the seats in that row.

I would look to optimize this by finding the seats for the largest blocks of passengers first as it would be easier to place single seats after all the bigger blocks have been filled.

### Concerns for Implementation

One implementation to help optimize this algorithm would be to avoid assigning any economy plus seats to the middle seat of a row. I noticed in running the original program that oftentimes there would be economy plus passengers put in the middle of a row with an empty seat on either side. If a window seat is not available or the customer did not specify wanting a window seat, I would look to automatically assign them to the next seat in from either window seat.

Something to consider when collecting information on ticket purchases would be if a group purchasing tickets would be willing to be seated separately if need be. While I would assume a group purchasing tickets would probably want to sit together, even one group willing to sit separately could help accommodate everyone else. This piece of information could be represented as a boolean value (***can\_seperate = true***). However, implementation of this feature could be difficult. It may require using a list for the values in the ***economy\_sold*** dictionary to store both the number of tickets and if they are willing to be seperated. It may also be necessary to specify if a group of 3 would want to be split into a group of 2 and 1 or all separated. This would apply if two parents were traveling with one young child and one parent could be separated or one parent was traveling with 2 older kids who could sit together (And the parent might be happy to have some alone time). This would also compound if we raised the ***max\_family\_size*** to 4. Finally, the seating for these blocks would need to happen after all the non-separated groups, so they would need to be pushed down to the end of the line in regards to seating order which may differ from their place in the dictionary. However, I do feel this would improve equity by offering options to regular economy passengers.

In some cases, after running ***fill\_plane*** there would be an empty seat. While this might be difficult to implement, it may be necessary to move an economy plus passenger in a non window seat to a different non window seat to help accommodate a block of economy seats.

I also noticed after running the program multiple times that sometimes, in some cases the flight would overbook it's seats and the remaining number of seats would be a negative number. This could be extremely inconvenient for groups looking to purchase seats. I would look to implement a feature in the purchasing of tickets that would not allow for a block of tickets to be sold if it exceeded the number of remaining seats. This may result in a few more empty seats on the plane, but that in turn could help with arranging blocks of tickets to sit together.

### **General Observations**

One thing that also stuck out to me about this program was that it was not necessarily realistic in terms of how a plane layout would be. In addition to window seats, aisle seats are also often sought after. I know I would usually prefer an aisle seat to a window seat on a longer flight. This would require a significant restructuring of the algorithm because it would involve keeping track of different passenger preferences (window, aisle, no preference) as well as seating of blocks of groups in regards to seating them across the aisle from one another. It would also depend on the seating structure (3 left, 3 right / 2 left, 3 center, 2 right etc)