allen_take_off_time

Marissa Allen

Overview:

The purpose of this project is to keep track of the information needed to schedule the airstrip resource for a queue of airplanes. As soon as airplane rolls out onto the taxiway they radio in their request to ground control and say that want to take off at a certain time slot. These requests are held in a file that is submitted to the allen_take_off_time program. This program will take care of the request identifier, request submission time, time slot requested, length of time requested, actual start time, and actual end time. It will do this by creating a class for all of the request information and combining it with a priority queue to sort the information. As the planes move along in their time slots the status of the queue will be printed out and a list will be printed of the plane's actual take off times.

Design:

This program contains four modules - allen_take_off_time, priority_queue, airstrip_schedule, and test_allen_take_off_time. The test_allen_take_off_time module is a unit test file to test any edge cases. For this program, the allen_take_off_time module will have a function called file_read to take in input from a file, read the request lines, separate the strings into a list containing one line per element, and load the file requests into the plane_requests function. The plane_requests function then passes the list of file requests into the class_requests function to be placed into the constructor of the class in the airstrip_schedule module. The file request information that the module will keep track of is the request identifier, request submission time, time slot requested, length of time requested, actual start time, actual end time. The plane_requests function will pass the unordered class list to the sorted_requests function, which loads all of this information into a priority queue using a subclass of the priority queue module. The priority queue is a list that has a seven-tuple inside of it. The priority queue establishes priority by using the request submission time as the highest priority value, the time slot requested as the next highest priority value, and the increase instance variable. The information stored in the priority queue is only ordered once it is dequeued. When the information is dequeued it returns a four-tuple of everything except the first three priority values used to establish order. The sorted_requests function appends these four-tuple values into a list. This list is passed once more through the class_requests function to generate a ordered list of class objects. The ordered list of class objects is passed to the print_format and final times functions. The print_format function shows the printed status of the queue as time moves forward; formatted to display the time slot values, plane identifier, and actual start time values from the list of class objects. Then the final_times function will print the actual listing of the take off times by formatting the plane identifier, actual start time, and actual end time values.