# Michael Chillemi

# 06/16/2023

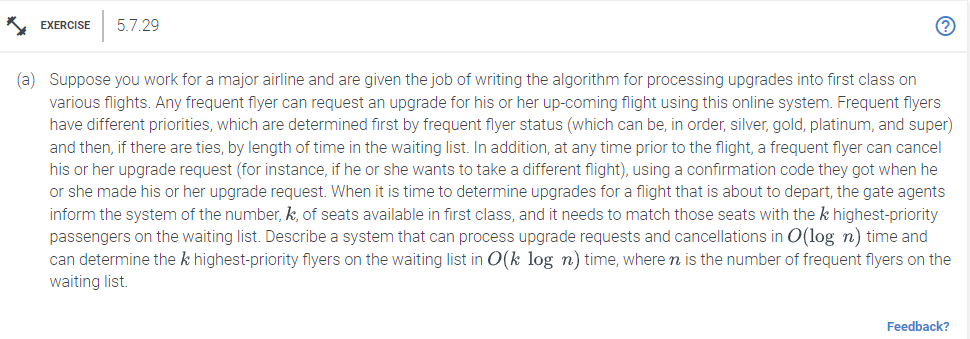
# CS 590 - Algorithms

# M4.B3: Module 4 Priority Queues and Heaps Application Exercises

# 

# 

Problem 5.7.29



An efficient system for dealing with this type of situation would be to store the frequent flyer data in a binary search tree data structure to process upgrade requests and cancellations in O(log n) time. The key of each node in the binary tree search will be the frequent flyer's priority level, and the value will be their confirmation code. We may add a new node to the Binary Search Tree when a frequent flyer seeks an upgrade, with their priority level serving as the key and their confirmation code serving as the value. We can find the node with the confirmation code and remove it from the Binary Search Tree in O(log n) time when a frequent flyer cancels their upgrade request. We can utilize a min-heap data structure to find the k highest-priority flyers on the waiting list in O(k log n) time. We can add the nodes to the min-heap after running the Binary Tree Search in order (from lowest priority to highest priority). We can eliminate the least priority node from the min-heap if its size exceeds k. The final k nodes in the min-heap will be the k highest-priority flyers on the waiting list once we have added all the nodes to the min-heap.This system will handle upgrade requests and cancellations in O(log n) time and find the k highest-priority flyers on the waiting list in O(k log n) time, where n is the number of frequent flyers on the waiting list.