CS577 Assignment 2

Due on Tuesday 6/26

1. Chapter 5 Question 3 in the textbook (Page 246-247). (*Hint: if there is one "dominant" account in the pile (i.e. more than half of the cards in the pile have the same account) , this account should be dominant in at least one of the half pile after dividing the big pile into two halves.*)

2. A special theater has only one row of n seats. Any time a patron is brought to his or her assigned seat, any patrons already seated next to this seat must rise and sit again, as must any next to them, and so forth. For ex ample, if there are patrons in seats 1, 2, 4, and 6, and a new patron is to be brought to sit in seat 3, the patrons in seats 2 and 4 must rise and sit again, which in turn requires the patron in seat 1 to rise and sit again; so seating the new arrival in this case causes three additional reseatings.
   As the owner of the theater, you would like to minimize the total number of seatings and reseatings necessary to seat the entire row of n patrons. Determine the order in which patrons should be brought to their seats so as to require only O(n log n) seatings and reseatings. Write a recurrence for the total number of (re)seatings and solve it. You may assume that n = 2^k -1 for some integer k > 0. (*Hint: Think about which seat should be saved for the last, and use recursion*)

3. A pair contains two numbers in an array with its second number is on the right side (not necessary immediately right) of the first one in an array. The difference of a pair is the minus result while subtracting the second number from the first one. Please :
   a. Use the divide and conquer methodology to design an efficient algorithm to find the *maximal difference* of all pairs in an array. For example, the maximal difference in the array {2, 4, 1, 16, 7, 5, 11, 3} is 13, which is the minus result of pair (16, 3).
   b. Analyze the computing complexity of your algorithm.