CSCI 570 - Fall 2016 - HW 3

1. Suppose you were to drive from USC to Santa Monica along I-10. Your gas tank, when full, holds enough gas to go p miles, and you have a map that contains the information on the distances between gas stations along the route. Let $d_1 < d_2 < ... < d_n$ be the locations of all the gas stations along the route where d_i is the distance from USC to the gas station (assume the distance between USC and Santa Monica is d_n). We assume that the distance between neighboring gas stations is at most p miles. Your goal is to make as few gas stops as possible along the way. Give the most efficient algorithm to determine at which gas stations you should stop and prove that your strategy yields an optimal solution. Give the time complexity of your algorithm as a function of n.

2. Solve Kleinberg and Tardos, Chapter 3, Exercise 3.

3. Solve Kleinberg and Tardos, Chapter 4, Exercise 4.

4. When we have two sorted lists of numbers in non-descending order, and we need to merge them into one sorted list, we can simply compare the first two elements of the lists, extract the smaller one and attach it to the end of the new list, and repeat until one of the two original lists become empty, then we attach the remaining numbers to the end of the new list and it's done. This takes linear time. Now, try to give an algorithm using $\mathcal{O}(n \log k)$ time to merge k ($k < n$) sorted lists (you can also assume that they contain numbers in non-descending order) into one sorted list, where n is the total number of elements in all the input lists. (Hint: Use a min-heap for k-way merging.)

5. Solve Kleinberg and Tardos, Chapter 4, Exercise 5.