Homework #2
(Course: CS 401)

These are the first two problems for Assignment 2. The deadline is November 13 (Tuesday), 2012, in class. No late assignments will be accepted.

The remaining problems of Assignment 2 will be given out later.

Special note: Any answer that is not sufficiently clear even after a reasonably careful reading will not be considered a correct answer, and only what is written in the answer will be used to verify accuracy. No hand waiving, vague descriptions or sufficiently ambiguous statements that can be interpreted in multiple ways will be considered as a correct answer, nor will the student be allowed to add any explanations to his/her answer after it has been submitted.

Problem 1 [**25 points**] There is a natural intuition that two nodes that are far apart in a communication network — separated by many hops — have a more tenuous connection than two nodes that are close together. There are a number of algorithmic results that are based to some extent on different ways of making this notion precies. Here is one way that involves the susceptibility of paths to the deletion of nodes.

Suppose that an n-node undirected graph $G = (V, E)$ contains two nodes s and t such that the distance beteween s and t is *strictly* greater than $^n/_2$. Let m be the number of edges of G.

(a) [**15 points**] Show that there must exist some node v, not equal to either s or t, such that the graph obtained from G by deleting v contains no path from s to t.

(b) [**10 points**] Give an algorithm with running time $O(m + n)$ to *find* such a node v as mentioned in **(a)** above.

Problem 2 (25 points): You have been working with some physicists who need to study, as part of their experimental design, the interactions among large numbers of very small charged particles. Basically, their setup works as follows. They have an inert lattice structure, and they use this for placing charged particles at regular spacing along a straight line. Thus we can model their structure as consisting of the points $\{1, 2, 3, \ldots, n\}$ on the real line; and at each of these points j, they have a particle with charge q_j. (Each charge can be either positive or negative.)

They want to study the total force on each particle, by measuring it and then comparing it to a computational prediction. This computational part is where they need your help. The total net force on particle j, by Coulomb's Law, is equal to

$$F_j = \sum_{i<j} \frac{C\, q_i\, q_j}{(j-i)^2} - \sum_{i>j} \frac{C\, q_i\, q_j}{(j-i)^2}$$

They have written the following simple program to compute F_j for all j:

for $j = 1, 2, ..., n$
  Initialize F_j to 0
  for $i = 1, 2, \ldots, n$
    if $i < j$ **then**
      add $\dfrac{C\, q_i\, q_j}{(j-i)^2}$ to F_j
    else if $i > j$ **then**
      add $\dfrac{-\,C\, q_i\, q_j}{(j-i)^2}$ to F_j
    endif
  endfor
  Output F_j
endfor

It is not hard to analyse the running time of this program: each invocation of the inner loop, over i, takes $O(n)$ time, and this inner loop is invoked $O(n)$ times total, so the overall time is $O\left(n^2\right)$.

The trouble is, for the large values of n they are working with, the program takes several minutes to run. On the other hand, their experimental setup is optimized so that they can throw down n particles, perform the measurements, and be ready to handle n more particles within a few seconds. So they would `really like it` if there were a way to compute all the forces F_j much more quickly, so as to keep up with the rate of the experiment.

Help them out by designing an algorithm that computes all the forces F_j in $O(n \log n)$ time.

HINT: How about finding two vectors whose convolution gives an entry

$$\sum_{i<j} \frac{q_i}{(j-i)^2} + \sum_{i>j} \frac{-q_i}{(j-i)^2}$$

for *every* j? For example, will the two vectors

$$(q_1, q_2, \ldots, q_n)$$

and

$$\left(\frac{1}{n^2}, \frac{1}{(n-1)^2}, \ldots, \frac{1}{4}, 1, 0, -1, -\frac{1}{4}, \ldots, -\frac{1}{(n-1)^2}, -\frac{1}{n^2}\right)$$

be OK?

After this, we can simply multiply each such term by Cq_j to get the final results.