Math 578A: Part 2, Homework 2

Due on May 1 Midnight

Total points: 80 points

Points will be given for correctness and clarity. You may submit a scan/photo of your work, however if it is difficult to read due to handwriting, etc., points may be deducted.

Question 1 *20 points*

Eulerian graphs

(a) [5 points] Write pseudo-code to determine if a graph has an Eulerian cycle.

(b) [15 points] Provide an algorithm to find an Eulerian cycle in an undirected graph in O(N) time.

Question 2 *20 points*

Assembly

(a) [10 points] An experiment attempts to provide the k-mer spectrum of a genome that does not have any repeats of length $k-1$, in either the forward or reverse direction, however the experiment randomly provides either a k-mer, or its reverse complement, for the entire spectrum (e.g. there are $n-k+1$ k-mers). Give an algorithm for constructing a genome.

(b) [10 points] Can one use a 12-inch-long wire to form a cube (each of the 12 cube edges is 1-inch long). If not, what is the smallest number of cuts one must make to form this cube?

Question 3 *20 points*

Approximation ratios

One of the first greedy algorithms you learned (but maybe didn't know it) is to make change. As input, you are given a list of coin values in decreasing value $C = c_1, c_2, \ldots, C_N$, and a value v, and you want to determine n the fewest coins to use to total v. For example, in US currency, $C = 25, 10, 5, 1$, and to give 0.82 you would give $3\ c_1, 0\ c_2, 1\ c_3, 2\ C_4$, $n = 6$. A greedy algorithm to find the number of coins to use is:

(a) [5 points] Provide a list of coin values and a total value for which GreedyChange incorrectly calculates the minimal number of coins to return as change.

(b) [5 points] Assuming there isn't a coin that exactly matches the return value, what is the minimal value for GreedyChange?

(c) [10 points] What is the approximation ratio for GreedyChange? Note this allows for any set of coin values C.

Question 4 *20 points*

Phylogenetics

Design a backtracking procedure to reconstruct the optimal assignment of characters in the Sankoff algorithm for the Weighted Small Parsimony problem.

Algorithm 1 Greedy change

procedure GreedyChange(C, v)
    $i = 1, n = 1$
    while $v > 0$ **do**
        $n_i \leftarrow \lfloor v/c_i \rfloor$
        $v \leftarrow v - n_i * c_i$
        $n \leftarrow n + n_i$
        $i = i + 1$
    end while
    return n
end procedure