CS172 Computability & Complexity (Spring'09)

Instructor: Mihai Pătraşcu GSI: Omid Etesami

Problem Set 8 *Due: Monday, April 13*

1. The Travelling Salesman Problem (TSP) is the following: Given a complete graph G with weighted edges, find a cycle visiting every node exactly once, such that the total length of the cycle is minimized. (You can imagine that the nodes are the destinations that a salesman wants to visit; he needs to visit all of them in some order, to minimize his total travel time.)

In this problem, you will solve the Planar TSP problem. The nodes will be points (x, y) in the plane; the distance between nodes (x_1, y_1) and (x_2, y_2) will be $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

Implement code that reads n and the coordinates of the n points (assume each is a floating point number). The code should output the length of the minimal cycle visiting all nodes.

Email the source code (in C, Java, or Pyhton) to `mip@alum.mit.edu` as an attachment to an email with the subject `CS172-HW8-P1`. Also email the answers to the following questions:

(a) What is the running time of your algorithm, using $O(\cdot)$ notation? (You will probably implement some backtracking algorithm.)

(b) If you generate $n = 10$ points randomly with coordinates in $[0, 1]$, what is the (estimated) average length of the tour? You should compute this estimation by running your code on, say, 100 random instances, and taking the average of the outputs.

(c) What is the largest n, such that your code typically finishes in under 10 seconds when run on instances of n random points?

2. In the Monotone TSP problem, the goal is to find the shortest cycle satisfying:

- the cycle starts at the Western-most point (minimum x).
- the cycle visits some nodes going only towards East (x increases at every step), until it reaches the Eastern-most point.
- the cycle visits other nodes going only West, until it returns to the Western-most point.
- each node is visited exactly once.

Assume that no two points have the same x coordinate (no two points are on the same longitude).

Implement an algorithm running in time $O(n^3)$ which solves the Monotone TSP problem on n points. Email the source code (in C, Java, or Pyhton) to `mip@alum.mit.edu` as an attachment to an email with the subject `CS172-HW8-P2`.

Also email the answers to the following questions:

(a) Briefly describe how your algorithm works. (Hint: use dynamic programming.)

(b) If you generate $n = 100$ points randomly with coordinates in $[0, 1]$, what is the (estimated) average length of the tour? You should compute this estimation by running your code on, say, 100 random instances, and taking the average of the outputs.

(c) What is the largest n, such that your code typically finishes in under 10 seconds when run on instances of n random points?

3. The TSP problem was not a language (the output was a number, not accept/reject). Here is a language for the so-called "decision version" of TSP:

$$L = \{(G, k) \mid \text{the graph } G \text{ has a cycle of length at most } k \text{ visting all nodes once}\}$$

In the language, the graph G is given by specifying the cost of every edge. For simplicity, assume that the cost of every edge is an integer between 1 and 1000.

(a) Prove that $L \in$ NP.

(b) Prove that if $L \in TIME(f(n))$, then the TSP problem (outputting the length of the shortest cycle) can be solved in time $O(f(n) \cdot \lg n)$.

4. Prove that the following language is in coNP:

$$L = \{(G, k) \mid \text{the longest simple path in } G \text{ has length at most } k\}$$

A simple path is a path that does not visit any node more than once.

5. We define 2HROTM to be 2-head Read-Only Turing Machines. These are Turing Machines with two heads that can be moved independently over the same tape. The machines are read-only, i.e. they may not write on the tape at all.

(a) Show that the halting problem for 2HROTM is decidable.

(b) Show that the emptiness problem for 2HROTM is undecidable.