
University of Colorado at Colorado Springs

Home Work Assignment 2

Due 10-06-2011

You do not need to hand in soft copies of your programs for this assignment. Hard copies, graphs, and written answers to questions are sufficient. I will ask you to demo if I have doubts about your results. If you email me your work, number your solutions by problem number, put all the files together (if you have several) in order, and create a single PDF before sending it to me. Print your name prominently. Please make it easy for me to figure out what's where.

1. Solve the following recurrence relations using any method you like. Assume that $T(n)$ is a constant for $n \leq 2$.

- (a) $T(n) = \sqrt{n} T(\sqrt{n}) + n$
- (b) $T(n) = 2T\left(\frac{n}{2}\right) + \frac{n}{\log n}$
- (c) $T(n) = 3T(n/2) + n \log n$

2. The Traveling Salesman Problem

- (a) Describe a brute-force solution to the traveling salesman problem. Write this algorithm clearly in pseudo-code.
- (b) Analyze the algorithm showing the steps clearly.
- (c) Implement the algorithm in a programming language of choice.
- (d) Randomly create n cities in 2-D space. Calculate distances between city pairs. Store the distances you generate in a file using a format you create. Consider using the format similar to the one you find in <http://people.sc.fsu.edu/~jburkardt/datasets/tsp/tsp.html>.
- (e) Run your TSP program on increasing number of cities by reading data from the file you generated in some order. Keep track of time needed. Graph the time needed against the number of cities and see if the times obtained experimentally agree with your theoretical results.
Keep track of the best solution to each of the problems generated in the same file where you store the problems themselves.
- (f) How many cities can you consider before your machine takes more than a certain amount of time to run, say 2 hours?
- (g) For the 10-city problem, draw the cities on a 2-D graph. Don't draw by hand; use a tool of some kind. Draw the best TSP tour among these ten cities.
- (h) Describe and write the pseudocode for a heuristic algorithm to solve the TSP problem. Analyze this algorithm. Run the algorithm on the datasets you have created earlier. Compare results when you can. You should be able to run the heuristic algorithm on much larger problems than the brute force method.
- (i) Run your heuristic algorithm with the datasets at <http://people.sc.fsu.edu/~jburkardt/datasets/tsp/tsp.html> and compare results with best results found at the Web site.