Special Assignment 2: Experimental Evaluation of Algorithms

CMSC-641 Algorithms, fall 2017, Alan Sherman, UMBC

Overview

Choose any problem you like (except matrix chain) that can be solved by dynamic programming. Experimentally address the question: *how much overhead, if any, does recursion incur in the solution based on dynamic programming to this problem?* Implement and experimentally measure the time and space complexity of two different implementations of your solution—iterative and recursive. The purpose of this assignment is to gain experience in measuring the complexity of algorithms experimentally, to gain insight through comparing theoretical and experimental running time and space usage, and to understand dynamic programming more deeply.

You may work in teams of up to three students each, and I recommend that you perform this work with the group for your research project (it is OK if this group has 2 or 4 students.) This assignment will help you get started on your experimental work for your project.

This project is intended to take about nine hours of time. Sharply focus your work on the aforementioned question and the requirements below. Focus sharply on your new and significant results.

Requirements

(a) Include two graphs: observed running times, and observed space usage. Each graph must include the results from both implementations on the same axis. Each graph must show observed means and standard deviations from several inputs for each of several selected input sizes. If reasonably possible, these input sizes should include at least the values 2^k, for k = 4, 5, …, 12. If needed, you may use a log-scale. If the standard deviations are too small to be noticeable, then note this fact.

(b) Using regression analysis (e.g., by Excel), match the observed data to the theoretical time and space analysis. That is, using regression analysis, find the best fit of coefficients to the theoretical time and space usage (do not simply find the best polynomial fit to the data, if the running time or space is not a polynomial). Do not fit some other arbitrary function (e.g., e^n) to the data, as many spread sheets will do unless otherwise instructed.

(c) Write up your results concisely in a short report (about 3-4 pages) with the following sections.

   (1) The problem and algorithms. What problem did you solve, and why is it important? State each of your two algorithms in high-level pseudocode. [about one paragraph plus pseudocode]

   (2) Methods. Briefly explain how you measured time and space. What language, system, and machine did you use? How many trials did you perform per input size? How did you generate your inputs? [about one paragraph]

   (3) Results. Present your two graphs. Briefly explain any notable features of these graphs. Every graph needs a number, name, and caption. Be sure also to label each axis with type (e.g., time), units (e.g., secs), and scale (e.g., 10^3). [about one paragraph per graph, plus two graphs]

   (4) Discussion. What do your data say about your research question? What are your recommendations for people who wish to use your algorithms? Focus sharply on your answer to the research question. [about one meaty paragraph]

   Write in the active (not passive) voice. Omit needless words.

Deliverables

Hand in on paper at beginning of class:

1. A succinct report
2. Source code.