Homework 5 – Due Friday, October 3, 2008

Please refer to the general information handout for the full homework policy and options.

Reminders

- Your solutions are due before the lecture. Late homework will not be accepted.
- Collaboration is permitted, but you must write the solutions *by yourself without assistance*, and be ready to explain them orally to a member of the course staff if asked. You must also identify your collaborators. *Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.*
- To facilitate grading, please write down your solution to each problem on a separate sheet of paper. Make sure to include all identifying information and your collaborators on each sheet. Your solutions to different problems will be graded separately, possibly by different people, and returned to you independently of each other.
- For problems that require you to provide an algorithm, you must give a precise description of the algorithm, **together with a proof of correctness** and an analysis of its running time. You may use algorithms from class as subroutines. You may also use any facts that we proved in class or from the book.

Problem to be handed in

Page limits: The answer to each problem should fit in **2 pages (or one double-sided sheet)** of paper. Longer answers will be penalized.

1. Given an array of temperatures $t_1, ..., t_n$, measured on n days, the weather service would like to compare the biggest increase in temperature to the biggest decrease in temperature over the given period. Specifically, you would like to compute two numbers: (a) max_{up}, the maximum over pairs $i < j$ of $t_j - t_i$ and (b) max_{down}, the maximum over pairs $i < j$ of $t_i - t_j$.

   For example, for the array $[10, 0, 1, 2, 3, 4, -1]$, you would compute $max_{up} = 4$ and $\max_{down} = 11$.

   Give a divide and conquer algorithm that computes these numbers in $O(n)$ time.

   Hint: Cut the array into two and make a recursive call on each half. It may be helpful to have the recursive calls pass up some extra information (in addition to $\max_{up}$ and $\max_{down}$).