Fall 2016: CS 430 - Quiz 7

**Time:** 35m

To help me grade, please respect the following guidelines for writing pseudocode:

1. C instructions are fine. But do not write object-oriented additions. Do not declare or use any class. Declare only procedures (if necessary) and explain in words what each procedure does, and what is the use of each parameter.

2. One instruction per line

3. from C, do not use ‘!’, ‘;’, ‘?’, or double assignment. As a general rule, this is not the place to be too smart

4. Match the brackets with a horizontal line

5. Number the lines of pseudocode

6. By default, the arrays are indexed 0…n – 1. If you prefer to use the range 1…n, it is OK, but you must write down this assumption.

7. It can help if you describe in English the idea of the algorithm. Examples of instances on which you run your algorithm do not help.

**Problem**

Given a sequence of numbers $a_1, a_2, \ldots, a_n$ (some of them might be negative) stored in an array, give an algorithm to find two indices $i \leq j$ such that $\sum_{k=i}^{j} a_k$ is maximum, among all the pairs $1 \leq i \leq j \leq n$. Your algorithm must run in $O(n)$ time.

**Hint:** Dynamic programming often computes and keeps track of more information than just the optimum solution for smaller problems. For example, to compute the longest path in a tree, the height of each subtree was also necessary. The difficult part, of course, is to find the right recursive function.

If we denote by $b_q = \max_{1 \leq i \leq j \leq q} \sum_{k=i}^{j} a_k$ (we are only interested in $b_n$), then there is no easy way to compute $b_{q+1}$ from $b_q$. So, a similar function $c_q$ is needed, with two properties:

- $c_q$ can be easily computed from $a_q$ and $c_{q-1}$
- The overall maximum can be easily computed from all $c_q$ (but it will not be $c_n$)

Don’t forget that indices $i$ and $j$ are needed, not the value of the summation. This can usually be easily accomplished by recording the best values for $i$ and $j$ whenever a bigger summation is found.