

Due at the start of class Friday, June 5, 2009.

Problem 1. Consider the following “brute force” solution for the maximum contiguous sum problem that adds up the values in every possible subsequence. The indices i and j are the starting and ending points of the subsequences.

```
M ← 0;
for i = 1 to n do
  for j = i to n do
    S ← 0;
    for k = i to j do
      S ← S + A[k];
    end for
    M ← max(M,S);
  end for
end for
```

- (a) Give a triple summation for the number of times the inner loop is executed.
- (b) Simplify the summation (a little bit) by getting a closed form for (just the) right most summation.
- (c) Simplify the summation (a little bit more) by getting a closed form for (just the) new right most summation. HINT: Use a “change of variable.”
- (d) Simplify the summation completely by getting a closed form for the entire summation. HINT: Use a “change of variable” and Problem 1a from homework 0.

Problem 2. Consider the problem of not only finding the value of the maximum contiguous sum in an array, but also determining the two endpoints. Give a linear time algorithm for solving this problem. [What happens if all entries are negative?]

Problem 3. We can generalize the “maximum contiguous sum problem” to two dimensions to solve the “maximum contiguous rectangle problem”. Given an $m \times n$ array of (positive and negative) numbers, find the largest sum of values in a (contiguous) rectangle.

- (a) Write down an English description of the “brute force” algorithm for the “maximum contiguous rectangle problem”. One or two sentences should suffice.
- (b) Write down the “brute force” algorithm in psuedocode.
- (c) How many times is the inner loop executed? Write it using summations.
- (d) Simplify your answer. Justify your work. [If you do this right, the solution involves very little calculation.]
- (e) **Challenge Problem.** Find a better algorithm for the maximum contiguous rectangle problem. How well can you do?