|  |  |
| --- | --- |
| **Project 1** | **CMPT 438 Algorithms** |
| **Due: Friday, December 9, 2016** | **Fall 2016** |

1. (**Programming Assignment**) Suppose you are given an array A with n entries, with each entry holding a distinct number. You are told that the sequence of values A[1], A[2], …, A[n] is unimodal: for some index p between 1 and n, the values in the array entries increase up to position p in A and then decrease the remainder of the way until position n. For example, the array A = [1 3 4 5 6 7 8 5 2 1] is unimodal, with the entries increasing up to position A[7] = 8 and decreasing afterwards. Implement an algorithm (in C++ or Java) that finds the “peak entry” p without having to read the entire array. Your algorithm should run in O(logn).

Submit a MS word document to Moodle (lms.manhattan.edu) that consists of the followings:

* A brief description of your algorithm and why its running time is O(logn)
* Source code
* Screenshot of the output