

COMP 250: Introduction to Computer Science

Assignment 1

Posted Friday, January 17, 2014
Due Wednesday, January 29, 2014

Please submit the homework through myCourses before midnight on the day it is due. You need to submit your .java files for the programming questions, and pdf files for the other questions (either typed or scanned).

1. [15 points] **More on list intersection**

Suppose that the arrays provided as arguments for listIntersection might contain duplicates. Write, in pseudocode, a list intersection algorithm for this case. You will have to return the number of *unique* elements in the intersection. E.g., for arrays:

1 3 2 3 5

and

1 3 4 1 7 3

your algorithm should return 2, because two distinct values (1 and 3) occur in both arrays.

2. [20 points] **A recursive algorithm problem**

A king owns a collection of n gold coins, stored in an array a . All coins have exactly the same weight except for exactly one that is actually filled with lead and is thus heavier than the normal gold coins. The king asks you to find the index of the lead coin as quickly as possible. You own a scale that allows you to compare the total weight of any two regions $a[i \dots j]$ and $a[k \dots l]$ of the array. Let us represent this scale by a method called $\text{compare}(a, i, j, k, l)$, defined as:

$$\text{compare}(a, i, j, k, l) = \begin{cases} -1, & \text{if } a[i \dots j] \text{ has weight smaller than } a[k \dots l] \\ +1, & \text{if } a[i \dots j] \text{ has weight greater than } a[k \dots l] \\ =, & \text{otherwise} \end{cases}$$

Using the compare method, write a recursive algorithm (in pseudocode) that returns the index of the lead coin. Your algorithm should make $O(\log n)$ calls to the compare method (but you don't need to prove this). Your algorithm has to work for any value of $n > 0$.

3. [25 points] **An array problem**

(a) [20 points] Suppose that you are given an array of integers. Write a Java program which prints out the element or elements that occur *least often* in the array, and the number of times they occur. For example, for array:

1 4 4 3 4 3 5 2 1

your algorithm should print the message:

Elements: 5 2

Number of occurrences: 1

The implementation should be done in the main method of your class.

(b) [5 points] What is the time complexity ($O()$) of your algorithm? Justify your answer.

4. [40 points] **Polynomials**

In this problem, you will extend the Poly class we are using in the lectures.

(a) [15 points] Write a method for multiplying two polynomials, with the signature:

Poly multiply(Poly p) throws Exception

Your method should throw an Exception if the argument p is null.

- (b) [5 points] Write a method which multiplies a polynomial by a constant. It should work by transforming the constant into a polynomial of degree 0, then calling the multiply method.
- (c) [10 points] Write a method that checks if there are any leading zeros in the coefficients of the polynomial. If so, it should re-size the coefficient array appropriately. Nothing is returned.
- (d) [10 points] Write a method that computes the derivative of a polynomial (and returns it as another polynomial)