CSE 331 - Summer 2015

PROGRAMMING ASSIGNMENT 1
Due Tuesday, June 16, 2015 @ 2:10pm

IMPORTANT: The stated deadline on assignments will be strictly enforced. I will go over solutions at the deadline while the problems are still fresh in your minds and will not accept submissions after the solutions have been presented.

Relevant code can be found at `https://bitbucket.org/hartloff/cse331-summer2015/` in the `assignments.assignment1` package. It is recommended that you clone the entire repository and pull regularly as all the code for this course will be placed there.

*You will receive no credit if you call a method that you haven't implemented yourself or is not implemented in the course repository (ie. calling `Math.pow` for question 2 will result in no credit). If you have doubt about a particular function call, ask about it on Piazza.

1 Polynomial evaluation

(50 points) For this question, you implement a method to perform integer polynomial evaluation in $O(d)$ time where d is the degree of the polynomial. This will be done by implementing Method 1. This method takes a polynomial defined by its coefficients and an evaluation value.

Method 1. `public static int polynomialEvaluation(int[] coefficients, int x);`

More formally, you are given a polynomial P defined by $coefficients = (a_0, \ldots, a_{d-1})$ for some integer $d \geq 1$ and an integer x such that $polynomialEvaluation(coefficients, x) = \sum_{i=0}^{n-1} a_i \cdot x^i$.

You goal for this problem is to implement Method 1 in $O(d)$ time (linear in the degree of P). No formal proof of runtime or correctness is required, though both will be verified during grading.

- (10 points) Implement an algorithm that correctly evaluates $P(x)$ for all inputs.
- (40 points) Implement an algorithm that is correct and runs in $O(d)$ time.

2 Exponetiation

(50 points) For this problem you will implement an algorithm that efficiently computes integer exponentiation. In particular, you will implement Method 2 which computes a^b. For full credit, your algorithm must run in $O(\log b)$ time. Note that $\log b$ is linear in the number of bits used to represent b so the algorithm will be linear in the size of b.

Method 2. *public static int exponentiation(int a, int b);*

- (10 points) Implement an algorithm that correctly evaluates a^b for all inputs.
- (40 points) Implement an algorithm that is correct and runs in $O(\log b)$ time.

3 Submission

Submit the file Assignment1.java which contains your implementation of the methods `polynomialEvaluation` and `exponentiation`.

The preferred method of submission is the `submit_cse331` command. I will also accept emailed submission. Keep in mind that I will be looking at the time stamp to ensure that the file was submitted before the deadline.