Martin Ziegler, Svetlana Selivanova *Total 10 pts*

Dongseong Seon, Hyunwoo Lee, Ivan Koswara, Seungwoo Schin

CS204

Fall 2018, Homework #6

Problem 1. *3 + 7 = 10pts*

In this problem, you should explain your algorithm using

- Operations that are given in the problem
- for-loop or while-loop, manipulating loop counter. You can always assume that loop counter increment is a primitive unit-time operation.
- if-else-elif statement
- return/print statement
- Declaration/assignment of a bit or arrays of bit variables
- Index operations on arrays of bits variables

If any other primitive operations or syntax is used, you should clearly state how it is defined. You will get **0 points** if an undefined operation or syntax is used without definition.
In a computer, integers are represented using the array of 0 and 1. In most cases, we can safely assume that summation of intergers are operations of constant time. This is because the number of bits required to represent a number is a logarithmic to the original number, so that the number of bitwise operations does not change a lot in usual occasions. However, when you think of extremely large numbers, number of bitwise operations should be considered. In this problem, you will design an addition of two integers using bitwise operations. Three bitwise operations are allowed in this problem; AND, OR, and NOT.

- AND : A binary operation with AND(0,0) = AND(0,1) = AND(1,0) = 0, AND(1,1) = 1.
- OR : A binary operation with OR(0,0) = 0, OR(0,1) = OR(1,0) = OR(1,1) = 1.
- NOT : A unary operation with NOT(0) = 1, NOT(1) = 0.

Now, using the bitwise operations above,

a) Design an algorithm add_1digit(p ,q) that accepts 1-bit input p, q and returns 2-bit output that contains binary representation of p+q.
b) Design an algorithm add(p, q) that accepts n-bit input p, q and returns (n+1)-bit output that contains binary representation of p+q. Analyze the asymptotic number of bitwise operations used.

Problem 2. *3 + 3 + 4 = 10pts*

Prove that

a) For any $f(n) \in \Theta(g(n))$ and $p(n) \in \Theta(q(n)) \rightarrow f(n) \cdot p(n) \in \Theta(g(n) \cdot q(n))$. Assume that f(n), g(n), p(n), q(n) are functions from positive integers to positive integers.
b) For $f(n) \in O(g(n))$ and $p(n) \in O(q(n))$, $f(n) + p(n) \in O(g(n) + q(n))$. Assume that f(n), g(n), p(n), q(n) are functions from positive integers to positive integers.
c) Order following functions in increasing order by their growth; $f(n) = ln(ln(n))$, $g(n) = (ln(n))^2$, $h(n) = ln(n!) - nln(n) + n$. In this problem, base of all logarithm functions are the natural constant (e = 2.71828...). (**Hint**: consider using Stirling's Formula)