**Problem 1:**

(1). for i = 1 to n { //loop executed n times

a = a + 1; //loop body takes c1

b = a + i; } // loop body takes c2

**Total running time, f(n) = n(c1 + c2),**

**F(n) = O(n)**

(2). for i = 1 to n { //Loop executed n times

for j = 1 to 2n //Loop executed 2n times

a = a / 2; } //Loop body takes c1

**Total running time: f(n) = n \* 2n(c1) = 2n2(c1),**

**F(n) = O(n2)**

(3). i = n

while (i >= 1) { // Loop executed Log(n) times, if loop variable multiplied or divided by constant.

a = a + 1; // Loop body takes c1

i = i / 2;} // Loop body takes c2

**Total running time: f(n) = log(n) \* (c1+c2),**

**F(n) = O(log(n))**

(4). i = n

while (i >= 1) { //Loop executed Log(n) times, if loop variable multiplied or divided by constant.

for j = 1 to n { // Loop executed n times

a = a + 1; } //Loop body takes c1

i = i / 2; } // Loop body takes c2

**Total running time: f(n) = (log(n)c2) \* n(c1) = nlog(n)(c1)(c2)**

**F(n) = O(nlog(n))**

(5). for i = 1 to n { // Loop executed n times

for j = 1 to n { // Loop executed n times

if j is even { // Loop body takes c1

a = a + 1; // Loop body takes c2

} } }

**Total running time: f(n) = n \* n(c1 + c2/2) = n2 (c1+c2/2),**

**F(n) = O(n2)**

**Problem 2:**

(1). Prove the following statement using the mathematical induction method that we discussed in the class: n2 > 3n for n ≥ 4

Base step: n = 4, LHS = 42 =16, RHS = 3\*4 = 12, LHS > RHS

Induction step: (n ≥ 4). Inductive hypothesis: it is true for n = k (k ≥ 4), i.e., k2 > 3k for k ≥ 4.

We show that it is also true for n = k + 1, i.e., (k+1)2 > 3(k + 1)

LHS = (k+1)2 = k2+2k+1

> 3k + 2k + 1 (by the inductive hypothesis k2>= 3k)

5k + 1 > 3k + 3

= RHS

So, LHS > RHS.

(2). Prove the following equation using the proof by contradiction method that we discussed in the class: If a number added to itself gives itself, then the number is 0.

Let x be a number of

X + X = X

For example, x = 1

1+1 = 1 which is wrong