CS 50010 Assignment 1

Due June 19th at start of class. Either bring a hard copy or email to bharsha@purdue.edu

Logic and Proofs

---

1) For each of the following statements, write the negation and contraposition
    a) If it is raining then I will bring my umbrella
    b) If I water my plants then they will grow
    c) For all real $x > 0$ there exists some real y such that $y^2 = x$ **(Negation only)**
    d) $(\neg P \vee Q) \rightarrow R$

2) Use a direct proof to show that $n^3 + n$ is even for every integer n.
3) Using induction, prove the following statement:

    For all natural numbers n, $\left(1+\frac{1}{1}\right)\left(1+\frac{1}{2}\right)\ldots\left(1+\frac{1}{n}\right) = n + 1$

Sets and Limits

For the following problems use these sets:

A = {1, 2, 3, 4, 5, 6}

B = {2, 3, 5, 7, 11, 13}

C = {π, e, $\sqrt{2}$, i }

1. Write the union and intersections for each of the following sets
    a. A and C
    b. A and B

c. B and C

2. Translate the following sets from English to “math” using set builder notation
    a. The set of all odd numbers
    b. The set of all perfect squares (Numbers that are equal to a squared natural number)
    c. The set of all real number **except** those between 5 and 10 (inclusive)

Asymptotics

For each of the following snippets of code, write the big O and Big Ω for the running time

1)

```
int foo(int n)
{
	sum = 0;
	for(int i = 0, i < n, i++)
	{
		sum = sum * 2;
	}
}
```

2)

```
void bar(int n)
{
	double temp = 1;
	if(n <= 0)
		return;
	for(int i = n, i > 0, i /= 2)
		temp *= 2;
}
```

3)

```
int foobar(int n, int m)
{
      int temp = 0;
      for(i = 1; i < n; i *=2)
            for(int j = 0; j < m; j++)
                  temp++;
      return temp;
}
```

Stacks, Queues, and Lists

---

1) Consider a stack implemented as an array with starting size 4. The stack is implemented with array doubling and shrinking. Draw the array and its contents as it would be as the following commands are run:
push(3), push(1), push(4), push(1), push(5), pop(),push(9),pop(),pop(),pop()

2) Complete the following table with the Big O notations of the runtimes (non amortized). Assume that resizing is implemented for array based implementations

	Array-Based	Linked List
Stack Push		
Stack pop		
Queue - Enqueue		
Queue - Dequeue		

3) Java's ArrayList class is implemented as a resizable array. However they do not use array doubling as we have talked about in class. Instead when the array is full they resize it to 1.5 times the current size. Show that this strategy also has constant amortized insertion cost.