CSC 382 Project 2

Average Case

In this project we will try to match the *Average Case* of algorithm A5 as we calculated in class and the "*Real Average*" of the algorithm.

```
A5: int Find (int x, int A[ ], int n) // array of size n
   { int j;
    for (j=0; j < n; j++) {
    (1)     if (x = = A[j]) {
              return (j+1); //the position is 1 more than the index
           }
    }
    return 0; // x is not an element of the array
  }
```

The project is composed of three steps:

1) (Calculated Average)

   a) Let $n = 50$.
   b) Let *bound* be an integer.
   c) Let x be an integer between *0* and *bound.*
   d) Let *hits=0.*
   e) Generate a sequence of n integers using a random number generator where the numbers of the sequence are between 0 and *bound.* Save this new sequence in an array *Sequence.*
   f) If x is equal to any of the numbers generated in the sequence increment *hits* by one (if x appears more than once *do not* increment *hits* every time).
   g) Repeat steps *e* and *f* 10,000 times (in other words create 10,000 sequences of 50 integers each. Clearly *Sequences* is a 50 x 10,000 two-dimensional array.
   h) Let *q=hits*/10,000.
   i) Calculate *A(n)* for algorithm A5 using the formula derived in class.

2) (Real Average)

Using the sequences created in the previous step (two-dimensional array *Sequences*) run algorithm A5 10,000 times using each time an input of *Sequences*. Let *total-steps* be a variable that tells you the total number of steps executed so far, that is, initialize *total-steps* to zero, and add the number of steps executed by A5 on each of the inputs, to this variable. For example if A5 executes 25 steps on the first input, then add 25 to *total-steps.* Let say on second input A5 performs 15 operations, now *total-steps* = 40 (i.e. 25 + 15). Keep doing this until all the 10,000 inputs are executed by A5. To calculate the Real Average let *A2*(*n*)=*total-steps*/10,000.

3) On step 3 compare *A*(*n*) as calculated in step 1) to *A2*(*n*) calculated in Step 2.

Run steps 1), 2) and 3) for the following values of *bound*

bound = 30, 50, 80, 100, 1000, 10,000, infinite

You final output should look like:

Bound	Calculated Average	Real Average
30	XX	XX
50	XX	XX
..	..	..
10,000	XX	XX
Infinite	XX	XX