CSE 2331 - Problem Set 2
Due beginning of lecture on September 24th

Problem numbers are from the third edition of "Introduction to algorithms". If unsure about which problem to solve, ask. Collaboration is permitted; looking for solutions from external sources (books, the web, etc.) is prohibited.

1. Give the asymptotic running time of the following algorithms in Θ notation. Justify your solution.

   (a)
   ```
   function func1(n)
   s = 0
   for i = n to n^2 do
       for j = i to n^2 do
           s = s + j - i
       endfor
   endfor
   return(s)
   ```

   (b)
   ```
   function func2(n)
   s = 0
   for i = 1 to n^2 do
       j = i^2
       while j > 10 do
           j = floor(j/5)
           s = s + j - i
       endwhile
   endfor
   return(s)
   ```

2. 4.4-2

3. Write a recurrence relation describing the worst case running time of each of the following algorithms and determine the asymptotic complexity of the function defined by the recurrence relation. Justify your solution. Assume that all arithmetic operations take constant time.

   (a)
   ```
   function func3(n)
   if n <= 10 then return(n)
   x = floor(n/7)
   x = x + func3(floor(3n/4))
   return(x)
   ```

   (b)
   ```
   function func4(A,n)
   /* A is an array of n integers */
   if n <= 2 then return (A[1])
   for i = 1 to floor(n/2) do
       A[i] = A[i] + A[i+1]
       B[i] = A[i] - A[i-1]
       C[i] = A[i] + A[2*i]
   endfor
   x = 0
   x = x + func4(A, floor(n/3))
   x = x + func4(B, floor(n/3))
   x = x + func4(C, floor(n/3))
   return(x)
   ```

4. Show that the following problem can be solved in time $O(n \log n)$: Given an array of n numbers, determine whether all the elements in the array are distinct.

5. Give the worst-case and expected case asymptotic running time of the following randomized algorithm in Θ notation. Justify your solution. `random(n)` generates a random integer between 1 and n with uniform distribution (every integer between 1 and n is equally likely).

```
function f(A[],n)
begin
    s = 0
    k = Random(n)
    for i = 1 to sqrt(k) do
```

```
        for j = 1 to i do
            s = s + A[i] * A[j]
        end
    end
    return(s)
end
```

6. 7.3-2