CSE 2331 Homework 5
Spring, 2016

For each of the following problems, simplify and express your answer as $\Theta(n^k)$ or $\Theta(n^k(\log n))$ wherever possible. If the asymptotic running time is exponential, then just give exponential lower bounds.

Random(n) generates a random number between 1 and n with uniform distribution (every integer between 1 and n is equally likely.)

1. Consider the following function:

```
Func1(A, n)
/* A is an array of integers                                   */
if (n ≤ 10) then return (0);
k ← Random(n − 1);
s ← 0;
for i ← 1 to n do
    for j ← 1 to n do
        for m ← 1 to n do
            A[i] ← A[i] + A[i + j + m];
            s ← s + A[i];
        end
    end
end
s ← s + Func1(A, k);
return (s);
```

   (a) What is the asymptotic worst case running time of `Func1`?

   (b) What is the asymptotic expected running time of `Func1`? Justify your solution.

2. Consider the following function:

```
Func2(A, n)
/* A is an array of at least j integers                        */
if (n ≤ 20) then  return (A[n]);
k ← Random(n − 1);
s ← Func2(A, k);
for i ← 1 to n − 1 do
    A[i] ← A[i] + A[i + 1];
end
s ← s + Func2(A, n-k);
return (s);
```

   (Note: Two recursive function calls to `Func2`.)

   (a) What is the asymptotic worst case running time of `Func2`?

   (b) What is the asymptotic expected running time of `Func2`? Justify your solution.

3. Let H be a hash table of size 7 with the hash function $h(K) = 4 * k \bmod 7$ implemented using CHAINED hashing. Consider the following sequence of insert operations:

   `Insert`(5, D_5);
   `Insert`(10, D_{10});
   `Insert`(12, D_{12});
   `Insert`(15, D_{15});
   `Insert`(4, D_4);
   `Insert`(40, D_{40});
   `Insert`(3, D_3);

   Draw the hash table after all the above operations have been executed, showing which data elements are in which locations of the hash table and how they are stored. Show your work.

4. Let H be a hash table of size 7 with the hash function $h(K, j) = (4 * k + 5 * j) \bmod 7$ implemented using OPEN ADDRESS hashing.

   Assume that $H[0], H[3], H[5]$ and $H[6]$ are already filled in the hash table so that the hash table looks like:

0	1	2	3	4	5	6
D_a			D_b		D_c	D_d

   Consider the following sequence of insert operations:

   Insert(3, D_e);
   Insert(6, D_f);

   Draw the hash table after all the above operations have been executed, showing which data elements are in which locations of the hash table. Show your work.

5. Let H be a hash table of size 55 with the hash function $h(K) = 15 * k \bmod 245$ implemented using CHAINED hashing. Explain why h is NOT a good hash function.

6. Consider the following problem:

   Let A be an array of records where each record contains three members:

   - A student id number;
   - A class section number;
   - The number of credit hours for the given class.

   Array A represents all the students and the classes they are taking in a given semester at OSU (Oregon State University).

   The total number of credit hours taken by a student with id number X is the sum of the credit hours of the classes taken by student X.

   Given array A and a positive integer h we wish to compute the total number of credit hours taken by each student and then report the NUMBER of students who are taking a total of h or more credit hours in the semester represented by A.

   (a) Write an algorithm (in pseudocode) which solves this problem USING A HASH TABLE. (Do NOT use sorting.) Your algorithm should use the hash table operations, Initialize, Insert, Retrieve and Member.

   Input to your algorithm is array A the size n of array A and a positive integer h. The algorithm returns the NUMBER of students who are taking h or more credit hours in the semester represented by A.

   (b) Assume your hash table has size at least $2n$. Analyze the EXPECTED asymptotic running time of your algorithm on an array.
   Justify your running time analysis.
   (You can assume either chained hashing or open address hashing.)

   (c) Assume your hash table has size at least $2n$. Analyze the WORST CASE running time of your algorithm on an array.
   Justify your running time analysis.
   (Note that the worst case time for insertion/find in a hash table is different than the expected case time. The difference between the worst case and expected times of insertion/find in the hash table will affect the worst case and expected times of your algorithm. You can assume either chained hashing or open address hashing.)