UNIVERSITY of TORONTO
SCARBOROUGH

Summer 2011 CSCB63
Assignment 1

Due Date: **June 14th, 9:00 a.m.**

Running Time

1) For each of the following pairs of functions, either $f(n)$ is in $O(g(n))$, $f(n)$ is in $\Omega(g(n))$, or $f(n) = \Theta(g(n))$. For each pair, determine which relationship is correct. Briefly explain your answer.

a. $f(n) = \log n^2$; $g(n) = \log n + 5$

b. $f(n) = n^{1/2}$; $g(n) = \log n^2$

c. $f(n) = \log^2 n$; $g(n) = \log n$

d. $f(n) = 2^n$; $g(n) = 3^n$

2) Using Big Oh notation, indicate the time requirement of each of the following tasks in the worst case. Describe any assumptions you make.

a. After arriving at a party, you shake hands with each person there.

b. Each person in a room shake hands with everyone else in the room.

c. You climb a flight of stairs.

d. You slide down the banister.

e. After entering an elevator, you press a button to choose a floor.

f. You ride the elevator from the ground floor to the nth floor

g. You read a book twice.

3) Using Big Oh notation, indicate the time requirement (worst case) of each of the following fragments of Java code. Describe any assumptions you make.

a.

```
a = b + c;
d = a + e;
```

b.

```
sum = 0;
for( i =0; i < 3; i++)
     for( j=0; j<n; j++)
          sum++;
```

c. Assume array A contains n values, `Random` takes constant time, and `sort` takes n log n steps.

```
for(i=0; i<n; i++){
     for(j=0; j<n; j++)
          A[i] = Random(n);
     sort(A , n);
}
```

d. Assume array `A` contains a random permutation of the values from 0 to n-1

```
sum3=0;
for(i=0; i<n; i++)
     for(j=0; A[j] != i; j++)
          sum3++;
```

4) For each of the following Java code fragments i) give an analysis of the running time (Big Oh), ii) implement the code in Java, and give the running time for several values of n, iii) compare your analysis with the actual running time. Include a tabular print out of the running times for different values of n you tried.

Notes: - make sure to choose a varying set of values for n from the small to the very large.

- To find the actual running time of a piece of Java code;

```
long lStart = System.currentTimeMillis();
// put code here
long lEnd = System.currentTimeMillis();
System.out.println(" time taken " + lEnd – lStart );
```

a.

```
sum = 0;
for(i=0 ; i < n; i++)
     for(j=0; j < n * n; j++)
          sum++;
```

b.
```
sum=0;
for(i=0; i < n; i++)
     for(j=0; j < i; j++)
          sum++;
```
c.
```
sum=0;
for(i=0 ; i < n; i++)
     for(j=0; j < i * i; j++)
          for(k=0; k < j; k++)
               sum++;
```

5) An Algorithm takes 0.5 ms for input size 100. How large a problem can be solved in 1 min if the running time is the following (assume the same hardware configuration is used in all cases):
a. linear
b. O(n log n)
c. quadratic
d. cubic

Binary Heap

6) A binary heap can either be implemented as a min-heap (pair with minimum key at root) or max-heap (pair with maximum key at root). This question is about max-heap implemented as a tree (not an array). Assuming we are going to insert the following in the same order form left to right:

```
10   5   12   3   2   1   8   7   9   4
```

Draw the resulting complete binary tree -- showing the insertions one at a time.

7) Assuming a binary heap is implemented using a tree data structure (not an array). Give an algorithm to find all nodes less than some value, X. Your algorithm should run in O(K), where K is the number of nodes output. Describe any assumptions you make.

Tree

8) Describe using pseudo code, a method that takes as input a binary search tree, T, and two keys k1 and k2, which are ordered so that K1 <= K2, and prints all elements X in the tree such that K1 <= Key(X) <= K2. Do not assume any information about the type of keys except that they can be ordered (consistently). Your method should run in O(K+log n) average time, where K is the number of keys printed.

Dictionary

9) Given input {4371, 1323, 6173, 4199, 4344, 9679, 1989} and a hash function h(x) = x (mod 10), show the resulting:
a. Separate chaining hash table.
b. Hash table using linear probing.
c. Hash table using quadratic probing.
d. Hash table with second hash function $h_2(x) = 7 - (x \bmod 7)$.

10) Show the result of inserting the following keys into an initially empty extensible hash table with 4 entries per bucket: 10111101, 00000010, 10011011, 10111110, 01111111, 01010001, 10010110, 00001011, 11001111, 10011110, 11011011, 00101011, 01100001, 11110000, 01101111.

Submission Instructions and Notes

- Assignment's solution must be done individually.

- Handwritten submission is acceptable BUT write legibly! YOU MUST WRITE IN PEN to be able to request remarking later.

- Stable your assignment solution together, number the pages, and attach a cover page with your name, UTORID, and student number.

- Late assignments is subject to 25% (absolute value) deducted for every day the assignment is late -- to a maximum of three days.

- There will be a 24 hour blackout prior to the due date - the TA is not going to be answering questions on the designated discussion board (http://portal.utoronto.ca) during that time.

- *Note:*
  - Computer Science department is currently moving to the new building (IC) Hence, the B63 drop-box location is yet to be determined. Once the information is available, it will be posted on the course announcement page (http://portal.utoronto.ca/)