Week 2: Lab

Collaboration level 0 (no restrictions). Open notes.

1. Consider the folowing code for BubbleSort that we discussed in class:

```
Bubble-Sort($A$)
1  For $k = 1$ to $n - 1$
2      // do a bubble pass
3      For $i = 0$ to $n - 2$
4          if $A[i] > A[i+1]$: swap
```

   (a) What is the best case of this algorithm?

   (b) Show how to change the code so that the algorithm has a best case of $\Theta(n)$ (Hint: does not do redundant bubble-passes).

   What we expect: Pseudocode of the new algorithm and best-case analysis.

2. Express the worst case running time of Insertion Sort as a sum.

3. Prove or disprove: $f = O(g)$ implies that $g = O(f)$.

   What we expect: If true, a justification that holds for any functions f, g and for all $n > n_0$. If false, a counterexample.

4. Prove or disprove: $f = O(g)$ implies that $g = \Omega(f)$.

   What we expect: If true, a justification that holds for any functions f, g and for all $n > n_0$. If false, a counterexample.

5. Find the order of growth of the following functions. For each function, give its $\Theta()$ and a justification.

   (a) $n \lg\lg n + n \lg n + \sqrt{n} \lg^2 n$

   (b) $\sqrt{n} \lg n + n$

   (c) $n^2 + \sqrt{n} \lg^3 n$

   (d) $3^{\lg n} + n^2 + n \lg n$

   (e) $\sqrt{3}^{\lg n} + n^2 + n \lg n$

   (f) $2^n + 2^{2n}$

   (g) $2^{\lg n} + \lg n^2$

(h) $(\lg n)^{\lg n} + n^3$

6. Arrange the following functions in ascending order of growth rate. For each pair of consecutive functions, give a brief justification on why they are in this order. For e.g., if you ordered A, B, C, you need to justify that 1. $A = O(B)$; and 2. $B = O(C)$.

$$2^{\sqrt{\log n}}, 2^n, n^{4/3}, n(\log n)^3, n^{\log n}, 2^{2^n}, 2^{n^2}$$

7. (interview question) You are given a set of n points on a circle in the plane. Come up with an algorithm that determines if there exists a pair of points that are antipodal (two points are antipodal if they are diametrically opposite). Analyze its running time.

8. (interview question) You are presented with 9 marbles. All of the marbles look identical i.e. same shape, color, and dimensions(except for weight). However, 8 of the 9 marbles have exactly the same weight; the last marble is heavier. The only tool you have to measure weights is an old fashioned balance scale. You are only allowed to use the scale 2 times. How do you find the one marble that is not the same weight as the others?

   Generalize to n marbles: find it using the scale $\log_3 n$ times.

9. Let $f(n) = \lg n$ and assume that we have an algorithm whose running time is $f(n)$ microseconds. Determine the largest size of a problem that can be solved by the algorithm in: (a) 1 second; (b) 1 hour; (c) 1 month; (d) 1 century.

   Same problem for $f(n) = n^{10}$ and $f(n) = 2^n$.

Optional

10. (interview question) Close all notes, pick a language of your choice, and implement: (a) bubble sort; (b) insertion sort; (c) binary search. Include tester functions (generate random arrays etc).

    Do not open your notes! The goal is to be able to go on your own from the high level description of the algorithms (which you should know), to the low level details, which you need to figure out on the spot (To make it more fun (or not), imagine someone is looking at your screen while you write).