# CS525DA
# HW#1

**DUE**: Tuesday, September 16 (Note that there are two pages)

1. (5 points) This result will be used when we get to NP-Completeness.
Design an algorithm to guess an unknown positive integer $n$ using $O(\log n)$ benchmark operations, where a benchmark operation is a comparison of the form

- $k > n$ ?
- $k = n$ ?
- $k < n$ ?

The values of $k$ will be determined by your algorithm. Another way to view this problem is to assume that there's an `oracle` who knows the value of $n$. The `oracle` will not tell you $n$'s value, but given any $k$ she will reply correctly to any of the benchmark operations. You may only ask the `oracle` at most $O(\log n)$ questions.

2. (1 point) Give a closed-form, as a function of $n$ and $m$, for $\sum_{n \geq k \geq m} k$ where $n \geq m \geq 0$.

3. (6 points) Prove or give a counter-example to each of the following:

A) **<u>CONJECTURE</u>**: If $f_1(n) \in O(g_1(n))$ and $f_2(n) \in O(g_2(n))$, then $f_1(n) + f_2(n) \in O(\max(g_1(n), g_2(n)))$.

B) **<u>CONJECTURE</u>**: If $f_1(n) \in O(g_1(n))$ and $f_2(n) \in O(g_2(n))$, then $f_1(n) + f_2(n) \in O(\min(g_1(n), g_2(n)))$.

C) **<u>CONJECTURE</u>**: If $f_1(n) \in \Theta(g_1(n))$ and $f_2(n) \in \Theta(g_2(n))$, then $f_1(n) + f_2(n) \in \Theta(\max(g_1(n), g_2(n)))$.

4. (4 points) If a problem $P$ has worst-case time complexity $\Omega(n \lg n)$ and worst-case time complexity $O(n^2)$ and algorithm $A$ solves problem $P$, which of the following are possible?

***a*** $A$ has best-case time complexity $\Theta(n)$.

***b*** $A$ has worst-case time complexity $\Theta(n\sqrt{n})$.

***c*** $A$ has average-case time complexity $\Theta(n^3)$.

***d*** $A$ has worst-case time complexity $\Theta(n)$.

5. (16 points) Do **Problem 4.1** on pg. 85 of our text.

# CS525DA
# HW#1 SOLUTIONS

1. $k \leftarrow 1$

**while** $k < n$ **do**

$k \leftarrow 2k$

/* $\frac{k}{2} \le n \le k$ */

**return** $Binary\_Search(k/2, k)$

*Binary_Search*(*lo*,*hi*)

$k \leftarrow \lfloor (lo + hi)/2 \rfloor$

**if** $k$=$n$ **then return** $k$

**if** $n > k$ **then return** *Binary_Search*($k$+1,$hi$)

**return** *Binary_Search*($lo$,$k$-1)

For the initial loop, we note that 1 will be doubled $\lceil \lg n \rceil$ times until it equals or exceeds $n$. Then *Binary_Search* will be executed $O(\lg n)$ times.

2. $\sum_{n \ge k \ge m} k = \sum_{n \ge k \ge 0} k - \sum_{m-1 \ge k \ge 0} k = \frac{n(n+1)}{2} - \frac{m(m-1)}{2}$

3. A) The CONJECTURE is true. Because $f_1(n) \in O(g_1(n))$ and $f_2(n) \in O(g_2(n))$, there exist $c_1$, $n_1$, $c_2$ and $n_2$ such that $f_1(n) < c_1 g_1(n)$ for all $n$>$n_1$, and $f_2(n) < c_2 g_2(n)$ for all $n$>$n_2$. Choosing $c^* = 2\max(c_1, c_2)$ and $n^* = \max(n_1, n_2)$ it follows that

$f_1(n) + f_2(n) < c_1 g_1(n) + c_2 g_2(n) \le \max(c_1, c_2)(g_1(n) + g_2(n))$

B) The CONJECTURE is false. Let $f_1(n) = g_1(n) = n$ and $f_2(n) = g_2(n) = n^2$. Then $f_1(n) + f_2(n) = n^2 + n$ and $\min(g_1(n), g_2(n)) = \min(n, n^2)$ =$n$. But $n^2 + n \notin O(n)$.

C) The CONJECTURE is true. Because $f_1(n) \in \Theta(g_1(n))$ and $f_2(n) \in \Theta(g_2(n))$, there exist $c_{1,1}$, $c_{1,2}$, $n_1$, $c_{2,1}$, $c_{2,2}$ and $n_2$ such that $c_{1,1} g_1(n) < f_1(n) < c_{1,2} g_1(n)$ for all $n$>$n_1$, and $c_{2,1} g_2(n) < f_2(n) < c_{2,2} g_2(n)$ for all $n$>$n_2$. Choosing $c_{lo} = \min(c_{1,1}, c_{2,1})$ and $c_{hi} = \max(c_{1,1}, c_{2,1})$, we add the two inequalities to get

$$c_{1,1}g_1(n)+c_{2,1}g_2(n)< f_1(n)+f_2(n)<c_{1,2}g_1(n)+c_{2,2}g_2(n)$$

$$c_{lo}2\min\left(g_1(n),g_2(n)\right)\le c_{lo}\left(g_1(n)+g_2(n)\right)< f_1(n)+f_2(n)<c_{hi}\left(g_1(n)+g_2(n)\right)\le c_{hi}2\max\left(g_1(n),g_2(n)\right)$$

So choosing $c_1=2c_{lo}$, $c_2=2c_{hi}$, and $n^*=\max(n_1,n_2)$, it follows that

$$f_1(n)+f_2(n)\in\Theta\left(\max\left(g_1(n),g_2(n)\right)\right).$$

4. ***a*** possible
   ***b*** possible
   ***c*** impossible
   ***d*** impossible - violation of worst-case time complexity $\Omega(n\lg n)$

5. ***a***. $a$=2, $b$=2, $f(n)$=$n^3$, $n^{\log_b a}=n^{\lg 2}=n$, $\frac{f(n)}{n^{\log_b a}}=\frac{n^3}{n}=n^2$ and case 3 of the Master Theorem applies, with $T(n)=\Theta(n^3)$.

***b***. $a$=1, $b$=10/9, $f(n)$=$n$, $n^{\log_b a}=n^{\log_{10/9}1}=1$, $\frac{f(n)}{n^{\log_b a}}=\frac{n}{1}=n$ and case 3 of the Master Theorem applies, with $T(n)=\Theta(n)$.

***c***. $a$=16, $b$=4, $f(n)$=$n^2$, $n^{\log_b a}=n^{\log_4 16}=n^2$, $\frac{f(n)}{n^{\log_b a}}=\frac{n^2}{n^2}=1$ and case 2 of the Master Theorem applies, with $T(n)=\Theta(n^2\lg n)$.

***d***. $a$=7, $b$=3, $f(n)$=$n^2$, $n^{\log_b a}=n^{\log_3 7}$, $\frac{f(n)}{n^{\log_b a}}=\frac{n^2}{n^{\log_3 7}}=n^{2-\log_3 7}$. Since 2>$\log_3$7, case 3 of the Master Theorem applies, with $T(n)=\Theta(n^2)$.

***e***. $a$=7, $b$=2, $f(n)$=$n^2$, $n^{\log_b a}=n^{\log_2 7}$, $\frac{f(n)}{n^{\log_b a}}=\frac{n^2}{n^{\lg 7}}=n^{2-\lg 7}$. Since 2<lg7, case 1 of the Master Theorem applies, with $T(n)=\Theta(n^{\lg 7})$.

***f***. $a$=2, $b$=4, $f(n)=\sqrt{n}$, $n^{\log_b a}=n^{\log_4 2}=\sqrt{n}$, $\frac{f(n)}{n^{\log_b a}}=\frac{\sqrt{n}}{\sqrt{n}}=1$. Case 2 of the Master Theorem applies, with $T(n)=\Theta(\sqrt{n}\lg n)$.

***g***. By unfolding,

$$T(n)=n+T(n-1)=n+(n-1)+T(n-2)=n+(n-1)+\ldots+2+T(1)=\sum_{k=2}^{n}k+\Theta(1)$$

$$=\sum_{k=1}^{n}k-1+\Theta(1)=\frac{n(n+1)}{2}-1+\Theta(1)=\Theta(n^2).$$

***h***. By unfolding, $T(n) = 1 + T(n^{1/2}) = 1 + 1 + T(n^{1/4}) = 1 + 1 + 1 + T(n^{1/8}) = \ldots = k + T\left(n^{2^{-k}}\right)$.

The iteration stops when $n^{2^{-k}} = 2$. Taking the lg of both sides, the iteration stops when $\frac{1}{2^k} \lg n = \lg 2 = 1$, or lg$n$=2$^k$, or $k$=lglg$n$. Thus, $T(n) = \lg \lg n$.