Homework 0

Due August 29, 2016
in lecture and SVN
Instructions for submission into your
class SVN repository are on the webpage.

The purpose of this assignment is to give you a chance to refresh the math skills we expect you to have learned in prior classes. These particular skills will be essential to mastery of CS225, and we are unlikely to take much class time reminding you how to solve similar problems. Though you are not required to work independently on this assignment, we encourage you to do so because we think it may help you diagnose and remedy some things you might otherwise find difficult later on in the course. If this homework is difficult, please consider completing the discrete math requirement (CS173 or MATH 213) before taking CS225.

Name: Aahan Agrawal

NetID: agrawl10

Section (circle one):

Wednesday	7–9pm AYB		
Thursday	9–11am AYC	11–1pm AYD	1–3pm AYE
	3–5pm AYF	5–7pm AYG	7–9pm AYH
Friday	9–11am AYI	1–3pm AYK	3–5pm AYL
	5–7pm AYM		
Laptop Sections:			
Thursday	9–11am AYS	11–1pm AYN	1–3pm AYO
	3–5pm AYP	5–7pm AYT	
Friday	9–11am AYU	1–3pm AYQ	3–5pm AYR
	5–7pm AYV		

Grade		Out of 60
Grader		

1. (3 points) Using 140 characters or less, post a synopsis of your favorite movie to the course piazza space under the "HW0 tell me something!" notice, so that your post is visible to everyone in the class, and tagged by #HW0num1. Also, use Piazza's code-formatting tools to write a *private* post to course staff that includes at least 5 lines of code. It can be code of your own or from a favorite project—it doesn't even have to be syntactically correct—but it must be formatted as a code block in your post, and also include the tag #HW0num1. (Hint: Check http://support.piazza.com/customer/portal/articles/1774756-code-blocking). Finally, please write the 2 post numbers corresponding to your posts here:

Favorite Movie Post (Public) number:	1015
Formatted Code Post (Private) number:	1182

2. (12 points) Simplify the following expressions as much as possible, **without using an calculator (either hardware or software)**. Do not approximate. Express all rational numbers as improper fractions. Show your work in the space provided, and write your answer in the box provided.

(a) $\prod_{k=2}^{n}\left(1-\frac{1}{k^2}\right)$

Answer for (a):	$\frac{n+1}{2n}$

$\prod_{k=2}^{n}\left[\frac{k^2-1}{k^2}\right]$

$\prod_{k=2}^{n}\left[\frac{(k+1)(k-1)}{k^2}\right]$

$= \frac{(2-1)(2+1)}{2^2} \cdot \frac{(3-1)(3+1)}{3^2} \cdot \ldots \frac{(k-1)(k+1)}{k^2} \cdot \frac{((k+1)-1)((k+1)+1)}{(k+1)^2} \cdot \ldots \frac{(n-1)(n+1)}{n^2}$

The cancellations leave only $\frac{1}{2} \cdot \frac{n+1}{n}$

$= \frac{n+1}{2n}$

(b) $3^{1000} \bmod 7$

Answer for (b):	4

$[3^6]_7 = 1$

since $\frac{3^6}{7} = 104 \text{ R } 1$

$[3^{1000}]_7 = [3^6]_7^{166} \cdot [3^4]_7$

$[3^{1000}]_7 = [1]_7 \cdot [4]_7$

$[4]_7$

(c) $\sum_{r=1}^{\infty}(\frac{1}{2})^r$

Answer for (c):	1

$\frac{a_0}{1-r}$

$\frac{\left(\frac{1}{2}\right)^1}{1-\frac{1}{2}}$ $=1$

(d) $\frac{\log_7 81}{\log_7 9}$

Answer for (d):	2

$\log_a b = \frac{\log_c b}{\log_c a}$

$=2$

$\log_9 81 =$ expression above

(e) $\log_2 4^{2n}$

Answer for (e):	4n

$\log_2 (2^2)^{2n} = \log_2 2^{4n} = 4n$

(f) $\log_{17} 221 - \log_{17} 13$

Answer for (f):	1

$\log_{17}\left(\frac{221}{13}\right)$

$= \log_{17}(17) = 1$

3. (8 points) Find the formula for $1+\underbrace{\sum_{j=1}^{n} j!j}_{f(n)}$, and show work proving the formula is correct using induction.

Formula:	$(n+2)!$

I claim the formula is $(n+1)!$

We prove this using induction

For $n=1$, $1+\sum_{j=1}^{n} j!j = 2 = (n+1)!$ ✓

Assume the formula is true for all n until $n=n$. We wish to show it also holds for $n=n+1$.

$f(n+1) = 1+\sum_{j=1}^{n+1} j!j = f(n) + (n+1)!(n+1)$

$= (n+1)! + (n+1)!(n+1)$

$= (n+1)!(n+2)$

$= (n+2)!$

4. (8 points) Indicate for each of the following pairs of expressions $(f(n), g(n))$, whether $f(n)$ is O, Ω, or Θ of $g(n)$. Prove your answers to the first two items, but just GIVE an answer to the last two.

(a) $f(n) = 4^{\log_4 n}$ and $g(n) = 2n+1$.

Answer for (a):	$f(n) = \Theta(g(n))$

$f(n) = n$ $\quad g(n) = 2n+1$

I claim $f(n) = \Theta(g(n))$

Proof: $\rightarrow$ $\forall n \geq 3$, $c=1$, $n \leq (1)(2n+1)$ $\leftrightarrow$ $f(n) \leq g(n)$

$\forall n \geq 2$, $c=3$, $(2n+1) \leq (3)(n)$ $\leftrightarrow$ $g(n) \leq 3f(n)$

(b) $f(n) = n^2$ and $g(n) = (\sqrt{2})^{\log_2 n}$.

Answer for (b):	$f(n) = \Omega(g(n))$

$f(n) = n^2$ $\quad g(n) = 2^{\frac{1}{2}\log_2 n} = \sqrt{n}$

I claim $g(n) = O(f(n))$, or $f(n) = \Omega(g(n))$

Proof: $\forall n > 1$, $c=1$, $\sqrt{n} \leq (1)n$

We also refute the claim that $f(n) = O(g(n))$.

Suppose $f(n) = O(g(n))$. Then $\exists n_0, c$, $\forall n \geq n_0$, $n^2 \leq c\sqrt{n}$

i.e. $n^{3/2} \leq c$ $\forall n \geq n_0$

This is a contradiction so $f(n) \neq O(g(n))$

(c) $f(n) = \log_2(n!)$ and $g(n) = n \log_2 n$.

We compare $n!$ and n^n after exponentiating each portion with 2

Answer for (c):	$f(n) = O(g(n))$

(d) $f(n) = n^k$ and $g(n) = c^n$ where k and c are constants and $c > 1$.

Compare n^5 and 5^n

Answer for (d):	$f(n) = O(g(n))$

5. (9 points) Solve the following recurrence relations for integer n. If no solution exists, please explain the result.

(a) $T(n) = T(\frac{n}{2}) + 5$, $T(1) = 1$; assume n is a power of 2.

$= T(\frac{n}{4}) + 5 + 5$ $= \dots$ after some more iterations

$= T(\frac{n}{8}) + 5 + 5 + 5$ $T(1) + (\log_2 n)5$

Suppose $n = 2^k$

Then $\log_2 n = k$

Answer for (a):	$5(\log_2 n) + 1$

(b) $T(n) = T(n-1) + \frac{1}{n}$, $T(0) = 0$.

Answer for (b):	$\sum_{k=1}^{n} \frac{1}{k} \Rightarrow$ no solution

$= T(n-2) + \frac{1}{n-1} + \frac{1}{n}$

after some iterations

$= \dots$ $\sum_{k=1}^{n} \frac{1}{k}$

There is no closed form for partial sums of the harmonic series.

(c) Prove that your answer to part (a) is correct using induction.

Recall that n is limited to $= 2^k$ where k is a non-negative integer

We induct on k. For $k=0$, $T(2^0) = T(1) = 1 \stackrel{\checkmark}{=} \underbrace{5(\log_2 1) + 1}_{\text{as per formula}}$

Assume this formula holds for all non-negative integers up to k. We prove it holds for $k+1$

$T(2^{k+1}) = T(\frac{2^{k+1}}{2}) + 5 = T(2^k) + 5 = 5(\log_2 2^k) + 1 + 5 = 5k+\dots$ $= 5(\log_2 \dots$

6. (10 points) Suppose function call parameter passing costs constant time, independent of the size of the structure being passed.

   (a) Give a recurrence for worst case running time of the recursive Binary Search function in terms of n, the size of the search array. Assume n is a power of 2. Solve the recurrence.

$T(n) = T(\frac{n}{2}) + C$

$T(1) = K$

Suppose $n = 2^k$

Then $\log_2 n = k$, so we have

$T(n) = \underbrace{T(1)}_{\text{after } \log_2 n \text{ divisions}} + c(\log_2 n)$

$= K + c(\log_2 n)$

Recurrence:	$T(n) = T(\frac{n}{2}) + C$
Base case:	$T(1) = K$
Recurrence Solution:	$K + c(\log_2 n)$

   (b) Give a recurrence for worst case running time of the recursive Merge Sort function in terms of n, the size of the array being sorted. Solve the recurrence.

A piazza post lets assume $n = 2^k$ for some $k \geq 0$

$T(n) = 2T(\frac{n}{2}) + n$

$T(1) = K$

Suppose $n = 2^k$, so

$\log_2 n = k$

Then $T(n) = \underbrace{2 \cdot 2 \cdot 2 \ldots}_{\log_2 n \text{ times}} + n(\log_2 n)$

$= 2^{\log_2 n} + n(\log_2 n)$

$= n + n(\log_2 n)$

Recurrence:	$T(n) = 2T(\frac{n}{2}) + n$
Base case:	$T(1) = K$
Running Time:	$n\log_2 n + n = O(n \log n)$

16→8→4→2→1→0

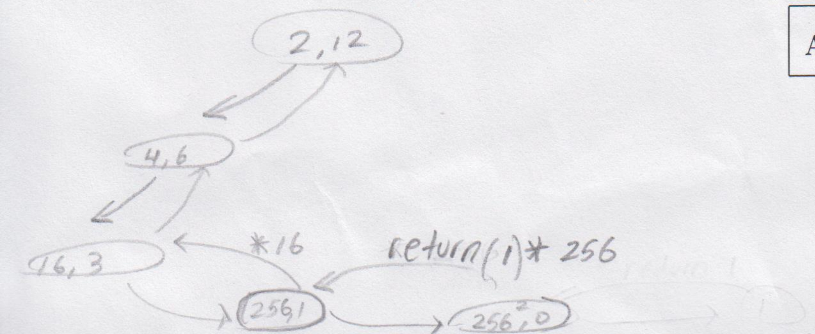
12→6→3→1→0

7. (10 points) Consider the pseudocode function below.

```
derp(x, n)
     if (n == 0)
         return 1;
     if (n % 2 == 0)
         return derp(x^2, n/2);
     return x * derp(x^2, (n-1)/2);
```

(a) What is the output when passed the following parameters: $x = 2$, $n = 12$? Show your work (activation diagram or similar).

Answer for (a):	16^3



(b) Briefly describe what this function is doing.

This function returns the value of x^n

(c) Write a recurrence that models the running time of this function. Assume checks, returns, and arithmetic are constant time, but be sure to evaluate all function calls. *Hint: what is the most n could be at each level of the recurrence?*

$T(n) = T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + k, \quad T(1) = C + 1$ ← This is the base relation, since the function must make one more call to $T(0)$

⇓ assuming worstcase: power of 2 = n

$\boxed{T(n) = T\left(\frac{n}{2}\right) + k \qquad T(1) = C + 1}$

(d) Solve the above recurrence for the running time of this function.

$T(n) = T\left(\frac{n}{2^q}\right) + \underbrace{k}_{q \text{ times}}$

Eventually

$= T(1) + \lfloor \log_2 n \rfloor k$ $\quad \lfloor \; \rfloor$ is the floor function

$= C + k \lfloor \log_2 n \rfloor + 1$