

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:

NetID:

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:	1124
Formatted Code Post (Private) number:	1249

2. (12 points) Simplify the following expressions as much as possible, **without using a 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):

$(n+1)/(2n)$

product sum $(k^2-1)/k^2 = \text{product sum } (k+1)(k-1)/k^2$
 $= \text{product sum } (k+1)/k * \text{product sum } (k-1)/k$
 product sum $(k+1)/k = 3/2 * 4/3 * 5/4 * \dots * (n+1)/n = (n+1)/2$
 product sum $(k-1)/k = 1/2 * 2/3 * \dots * (n-1)/n = 1/n$
 so result is $(n+1)/2n$

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

Answer for (b):

4

mod 7
 $3^0=1$
 $3^1=3$
 $3^2=2$
 $3^3=6$
 $3^4=4$
 $3^5=5$
 $3^6=1$
 $1000/6=166 \dots 4$
 so $3^{1000} \bmod 7$ is 4

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

Answer for (c):

1

$$S(r) = \text{sum of } (1/2)^r \text{ (r=1 to infinity)} = 1/2 + 1/4 + 1/8 + \dots + (1/2)^r$$

$$2S(r) = 1 + 1/2 + 1/4 + \dots + (1/2)^{(r-1)} = 1 + S(r) - (1/2)^r$$

$$S(r) = 1 - (1/2)^r$$

when r is infinity, $(1/2)^r = 0$

so the answer is 1

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

Answer for (d):

2

$$= \log_9 81 = 2$$

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

Answer for (e):

4n

$$= 2n \log_2 4 = 2n * 2 = 4n$$

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

Answer for (f):

1

$$= \log_{17} (221/13) = \log_{17} 17 = 1$$

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

Formula:	$(n+1)!$
----------	----------

$$\begin{aligned}
 S(n) &= 1 + \sum_{j=1}^n j! \quad (j=1 \text{ to } n) \\
 \text{base case: } S(1) &= 1 + 1! = 2 \\
 \text{induction part: } S(n+1) &= 1 + \sum_{j=1}^{n+1} j! = (n+2)! \\
 &= \sum_{j=1}^n j! + (n+1)! + 1 \\
 &= (n+1)! + (n+1)! + 1 \\
 &= (n+1)!(n+2) + 1 \\
 &= (n+2)!
 \end{aligned}$$

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):	$f(n) \Theta g(n)$
-----------------	--------------------


$$\begin{aligned}
 f(n) &= n \\
 n &< C(2n+1) \text{ for } n > 1, C > 0
 \end{aligned}$$

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


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

$$\begin{aligned}
 g(n) &= n^{1/2} \\
 n^2 &> Cn^{1/2} \text{ for } C > 0, n > 1
 \end{aligned}$$

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

Answer for (c):	$f(n)$  $g(n)$
-----------------	---

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

Answer for (d):	$f(n)$  $g(n)$
-----------------	---

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.

$$\begin{aligned} T(n) &= T(n/2^i) + 5i \\ n/2^i &= 1 \\ n &= 2^i \\ i &= \log_2 n \\ T(n) &= 5 \log_2 n + 1 \end{aligned}$$

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

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

Answer for (b):	Harmonic series
-----------------	-----------------

$$T(n) = T(n-1) + 1/n = T(n-2) + 1/(n-1) + 1/n = \dots = 0 + 1 + 1/2 + 1/3 + \dots + 1/n$$

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

$$\begin{aligned} \text{base case: } T(1) &= 1 = 0 + 1 = 1 \\ \text{induction part: } T(2^{k+1}) &= 5 \log_2(2^{k+1}) + 1 = T(2^k) + 5 = \\ &= 5 \log_2(2^k) + 1 + 5 = 5(k+1) \log_2 2 + 1 = 5 \log_2(2^{k+1}) + 1 \end{aligned}$$

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.

Tree height $= \log_2 n$

$$T(1) = C_1$$

$$T(n) = T(n/2) + C_2$$

$$= T(n/2^i) + iC_2$$

$$= C_1 + C \log_2 n$$

Recurrence:	$T(n/2) + C_2$
Base case:	C_1
Recurrence Solution:	$C \log_2 n + C_1$

- (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.

Tree height $= \log_2 n$

$$T(1) = C_1$$

$$T(n) = 2T(n/2) + nC_2 + C_3$$

$$C_i = \log_2 n = 2^i T(n/2^i) + (2i+1)n + (2i+1)C_2$$

$$T(n) = Cn \log_2 n + d$$

Recurrence:	$2T(n/2) + nC_2 + C_3$
Base case:	C_1
Running Time:	$O(n \log_2 n)$

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):	4096
-----------------	------

$(x=2, n=12) \rightarrow (x=4, n=6) \rightarrow (x=16, n=3) \rightarrow (x=256, n=1) \rightarrow 16 \cdot 256 = 4096$

- (b) Briefly describe what this function is doing.

Computing $f(x)=x^n$

- (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?*

$$\begin{aligned}
 T(0) &= d \\
 T(n) &= T(\lfloor n/2 \rfloor) + C
 \end{aligned}$$

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

$$T(n) = C \log_2 n + d$$

Blank sheet for extra work.