

CS 4349.501 Homework 1

Due Wednesday September 6th, in class

August 28, 2017

Please answer each of the following questions. Each student must write their solutions in their own words and submit their solutions on paper at the beginning of class. ***Include your name and/or Net ID at the top of each page.*** In this and future homework assignments, it may help to recall the harmonic numbers $H_n = 1 + 1/2 + \dots + 1/n = \Theta(\log n)$ and the basic log definitions $\log_2 n = \lg n \neq \ln n = \log_e n$.

Some important course policies

- Unless the problem states otherwise, you must prove your solutions are correct. Generally, when we ask you to “design and analyze an algorithm”, then you must:
 1. describe the algorithm using an unambiguous set of instructions (a mix of English explanation and pseudocode is probably a good idea),
 2. prove the algorithm is correct, and
 3. give an asymptotic bound for its running time along with a brief explanation for the bound.
- We strongly suggest you use \LaTeX to typeset your solutions. Any illegible solutions will be considered incorrect. The announcement for this homework links to a template for writing solutions in \LaTeX .
- If you do use outside sources or write solutions in close collaboration with another student, then you may cite that source or student and still receive full credit for the solution. Material from the lecture, the required textbook, or prerequisite courses need not be cited. Failure to cite other sources or failure to provide solutions in your own words, even if quoting a source, is considered an act of academic dishonesty.

See <https://utdallas.edu/~kyle.fox/courses/cs4349fa17/about.shtml> for more detailed policies and some advice. If you have any questions about these policies, please do not hesitate to ask in class, in office hours, or through email.

1. (**Erickson Appendix II.**) A *tournament* is a directed graph with exactly one directed edge between each pair of vertices. That is, for any vertices v and w , a tournament contains either an edge $v \rightarrow w$ or an edge $w \rightarrow v$, but not both. A *Hamiltonian path* in a directed graph G is a directed path that visits every vertex of G exactly once.

Prove that every tournament contains a Hamiltonian path. **Hint:** Use strong induction.

2. Using Θ -notation, provide asymptotically tight bounds in terms of n that answer each of the following questions. For example, if asked for $\sqrt[3]{n^3}$, then your answer should be $\Theta(n)$. **Do not turn in proofs for this problem.**

- What is $\sum_{i=1}^n i^2$?
- What is $\sum_{i=1}^n \frac{i}{n}$?
- What is $\sum_{i=1}^n \frac{n}{i}$?
- What is $n^2/50 + \sqrt{n} + 1/n$?
- What is $2^{(1/2)\lg n}$?
- What is the worst-case time to find an item in a balanced binary search tree with n nodes?
- What is the number of bits needed to write 10^n in binary?
- Let n be any natural number. Suppose the number of students at a well-known Texas university doubles every ten years. If this university has 27,000 students at the beginning of 2017, how many students will it have in the year 2117?
- What is the worst-case running time of the DoSOMETHING algorithm given below?
- What is the best-case running time of DoSOMETHING?

```

DoSOMETHING(A[1 .. n]):
  for i ← 1 to n - 1
    j ← i + 1
    while j ≤ n and A[j] < A[i]
      for k ← j to n
        A[i] ← A[k]
      A[i] ← A[j]
      j ← j + 1
  output A

```

3. Sort the functions listed below from asymptotically smallest to asymptotically largest, indicating ties if there are any. **Do not turn in proofs for this problem.** To simplify your answers, write $f(n) \ll g(n)$ to mean $f(n) = o(g(n))$, and write $f(n) \equiv g(n)$ to mean $f(n) = \Theta(g(n))$. For example, the functions n , n^2 , $2n^2$, and n^3 could be sorted as $n \ll 2n^2 \equiv n^2 \ll n^3$.

$500n^2$	$\lg^2 n$	1.001^n	20	$2n^{500}$	$\lg \lg n$	$H_{\sqrt{n}}$	n
$n^4 - (n-1)^4$	$2 + \sin n$	$\lg^{\sqrt{2}} n$	$\sqrt{\lg n}$	2^n	n^2	$\lg n$	H_{2^n}
$\ln(5n)$	$n\sqrt{n}$	e^n	$4^{\lg n}$	\sqrt{n}	$n^{1/1000}$	$n \lg n$	$\lg \lg \lg n$

4. You've been asked to throw a party for the newly formed UTD Pet Alliance, an uneasy truce between the university's cat lovers, dog lovers, parrot lovers, etc. Everybody is very passionate and only loves their one favorite kind of pet. In fact, you have heard that one animal in particular is loved by strictly more than half the guests, and you need to keep a close eye on them to make sure they do not taunt the other guests with more eccentric tastes. Unfortunately, it is rude to simply ask participants their favorite animal. You should just know; do they look like a _____ lover to you? All you can do is introduce pairs of guests; if they get along then they must love the same animal, if they turn their noses at each other, they must love different animals. How do you find this majority of guests that love the same animal?

More formally, you are given an array $X[1..n]$. The only method you have to compare elements of X is a procedure $\text{SAME}(x, y)$ that returns `TRUE` if elements x and y are equivalent and `FALSE` otherwise. Design and analyze an algorithm to output a member of X whose equivalence class contains strictly greater than $n/2$ members. For your analysis, give an asymptotic bound on the number of times your algorithm calls SAME .

Note: Expressing the number of calls to SAME as a recurrence is worth substantial partial credit, but you should be able to give an asymptotically tight answer by now.