

EECS 340
Algorithms and Data Structures
2005 Spring Semester

Homework 2

Due at the beginning of class, February 11, 2004

Read: Lectures 2, 3, 4, 7 and Supplemental Lecture 4.

1. The following questions ask you to prove mathematical facts by induction.
 - a. Find a formula for $\sum_{i=1}^n (-1)^{i-1} i^2$ by examining the first few values of this expression for small values of n . Then, prove the equality for arbitrary values of $n \geq 1$ using induction.
 - b. Prove by induction that $H_{2^n} \geq 1 + \frac{n}{2}$.
 - c. Find an expression for $T(n) = 4T\left(\left\lfloor \frac{n}{5} \right\rfloor\right) + 6n \log n$, $T(1)=0$ using the Master Theorem. Explain how you applied the Master Theorem. You can assume that $n \geq 1$. Conjecture an upper bound, which should be as tight as you can make it. The upper bound should be an exact formula (no big-Oh). Prove your upper bound for arbitrary values of n using mathematical induction. (*Hint:* proof in Supplemental Lecture 4.)
 - d. Prove that 6 divides $n^3 - n$ for all $n \geq 2$ by using mathematical induction.
 - e. Use induction to prove that there are $n!$ permutations (ordered arrangements) of the integers 1, 2, ..., n . Assume that $n \geq 1$.
 - f. Use induction to prove that radix sort works. Where does your proof need the assumption that the intermediate sort is stable? Make sure to state the quantity on which you do induction.
2. The following question is about the sieve method. Suppose you have two arrays A and B with n elements, none of which is repeated, sorted in ascending order. Describe an $O(\log n)$ algorithm that finds the k th smallest element in the union of A and B . The union of A and B are all the $2n$ distinct elements that appear in either A or B . Assume that there is no repeated element in the union of A and B . Explain why your algorithm is correct and takes $O(\log n)$ time in the worst case.

(over)

3. Suppose that we have a set of activities to schedule among a large number of lecture halls. We wish to schedule all the activities using as few lecture halls as possible. Give an $O(n \log n)$ algorithm to determine which activity should use which lecture hall.