

Due at the start of class Wednesday, June 27, 2007.

Problem 1. Consider an array of size eight with the numbers in the following order 10, 30, 50, 70, 20, 40, 60, 80.

- (a) What is the array after heap formation? How many comparisons does the standard algorithm use?
- (b) Show the array after each element sifts down *after heap creation*. How many comparisons does the standard algorithm use for all of the sifts?
- (c) How many comparisons does the modified algorithm (Floyd's version) use to create the heap?
- (d) How many comparisons does the modified algorithm (Floyd's version) use for the remainder of the sort?

Problem 2. We are going to find the exact number of comparisons for heap creation in a complete binary tree. We will guess the formula by looking at a few small examples, and then prove it is correct by mathematical induction.

- (a) Calculate by hand the exact number of comparisons for complete trees with 0, 1, 2, 3, 4 levels.
- (b) We know that the true answer should be approximately $2n$. Find the differences between $2n$ and your calculated values.
- (c) Guess a formula for your differences as a function of n .
- (d) What formula does that give you for the exact number of comparisons for heap creation as a function of n ?
- (e) Heap creation can be thought of as a recursive procedure: Create heap for left child of root, create heap for right child of root, and sift down root value. Write a recurrence for the number of comparisons to create a heap.
- (f) Use mathematical induction to prove that your formula is a solution to the recurrence.

Problem 3. Consider an array of size eight with the numbers 60, 80, 20, 10, 50, 30, 70, 40. Assume you execute quicksort using the version of partition from CLRS.

- (a) What is the array after the first partition. How many comparisons did you use? How many exchanges?
- (b) Show the left side after the next partition. How many comparisons did you use? How many exchanges?
- (c) Show the right side after the next partition on that side. How many comparisons did you use? How many exchanges?
- (d) What is the total number of comparisons in the entire algorithm? What is the total number of exchanges in the entire algorithm?

Problem 4. Assume you execute quicksort using the version of partition from CLRS.

- (a) What is the fewest comparisons that the algorithm will execute for an input of size 7.
- (b) Give an example of such an input (for $n = 7$).

Problem 5. Assume you execute quicksort using the version of partition from CLRS.

- (a) What is the fewest exchanges that the algorithm will execute for an input of size n .
- (b) Give an example of such an input for $n = 8$.