

Due at the start of class Wednesday, June 25, 2003.

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

- (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 derive a good upper bound on the worst case number of comparisons in (standard) heapsort. In order to simplify the algebra assume that the size of the list that you are sorting is one less than a power of two (i.e., a complete tree). Thus, an array of size $n = 2^k - 1$ is associated with a binary tree with k levels numbered $0, 1, \dots, k - 1$.

- (a) How many nodes are on level j ?
- (b) How many comparisons does the heap formation phase use to sift an element rooted on level j in the worst case?
- (c) Give a summation for the total number of comparisons for heap formation.
- (d) Simplify the summation.
- (e) How many comparisons does the remainder of heapsort use for each sift after removing an element from level j ? Note that the tree gets smaller so not all elements on level j have exactly the same number of comparisons.
- (f) Give a summation for the total number of comparisons for the second phase of heapsort.
- (g) Simplify the summation.
- (h) Add the two totals and simplify.

Problem 3. Consider an array of size eight with the numbers 50, 70, 10, 20, 60, 40, 80, 30. 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 $n = 8$.

Problem 6. Challenge problem. Assume n is a power of 2. If you view heap creation as a recursive procedure, you get approximately the following recurrence for the number of comparisons:

$$T(n) = \begin{cases} 2T(n/2) + 2 \lg n & \text{if } n > 1 \\ 0 & \text{if } n = 1 \end{cases}$$

- (a) Use constructive induction to show that $T(n) = an + b \lg n + c$. Find constants a , b , and c .
- (b) Solve the recurrence using the iteration method.