Problem Set 9

MIT students: This problem set is due in lecture on **Friday, December 9, 2005.** The homework lab for this problem set will be held 6–8 P.M. on Thursday, December 8, 2005.

Reading: Chapter 27 and Handout 29.

Both exercises and problems should be solved, but *only the problems* should be turned in. Exercises are intended to help you master the course material. Even though you should not turn in the exercise solutions, you are responsible for material covered in the exercises.

Mark the top of each sheet with your name, the course number, the problem number, your recitation section, the date and the names of any students with whom you collaborated. **Please staple and turn in your solutions on 3-hole punched paper**.

You will often be called upon to "give an algorithm" to solve a certain problem. Your write-up should take the form of a short essay. A topic paragraph should summarize the problem you are solving and what your results are. The body of the essay should provide the following:

1. A description of the algorithm in English and, if helpful, pseudo-code.
2. At least one worked example or diagram to show more precisely how your algorithm works.
3. A proof (or indication) of the correctness of the algorithm.
4. An analysis of the running time of the algorithm.

Remember, your goal is to communicate. Full credit will be given only to correct solutions *which are described clearly*. Convoluted and obtuse descriptions will receive low marks.

Exercise 9-1. Do Exercise 1-3 on page 6 of Handout 29.

Exercise 9-2. Do Exercise 1-4 on page 6 of Handout 29.

Exercise 9-3. Do Exercise 1-6 on page 7 of Handout 29.

Exercise 9-4. Do Exercise 2-4 on page 12 of Handout 29.

Exercise 9-5. Do Exercise 2-8 on page 13 of Handout 29.

Exercise 9-6. Do Exercise 27.1-5 on page 708 of CLRS.

Exercise 9-7. Do Exercise 27.2-2 on page 711 of CLRS.

Exercise 9-8. Do Exercise 27.3-6 on page 716 of CLRS.

Exercise 9-9. Do Exercise 27.4-2 on page 718 of CLRS.

Problem 9-1. More parallel merge sort

In this problem we will improve the parallel merge-sort algorithm from lecture. The algorithm described in class has work $\Theta(n \lg n)$ and parallelism $\Theta(n/\lg^2 n)$. We shall develop an algorithm with the same work, but higher parallelism.

(a) Given two sorted arrays containing a total of n elements, give an algorithm to find the median of the n elements in $\Theta(\lg n)$ time on one processor.

(b) Using the algorithm in part (a) as a subroutine, give a multithreaded algorithm to merge two sorted arrays. Your algorithm should have $\Theta(n)$ work and $\Theta(n/\lg^2 n)$ parallelism. Give and solve the recurrences for work and critical-path length, and show that the parallelism is $\Theta(n/\lg^2 n)$, as required.

(c) *Optional:* Generalize the algorithm in part (a) to find an arbitrary order statistic. Using this algorithm, describe a merge-sorting algorithm with $\Theta(n \lg n)$ work that achieves a parallelism of $\Theta(n/\lg n)$.