Problem Set 1

This problem set is due **in lecture** on **Monday, September 16.**

Reading: Chapters 1–4 (excluding §4.4); §28.2; §30.1

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 by 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.

Each problem should be done on a separate sheet (or sheets) of three-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 your essay should provide the following:

1. A description of the algorithm in English and, if helpful, pseudocode.
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. Graders will be instructed to take off points for convoluted and obtuse descriptions.

Exercise 1-1. Do Exercise 2.3-5 on page 37 in CLRS.

Exercise 1-2. Do Exercise 2.3-7 on page 37 in CLRS.

Exercise 1-3. Do Exercise 3.1-1 on page 50 in CLRS.

Exercise 1-4. Do Exercise 3.1-3 on page 50 in CLRS.

Exercise 1-5. Do Exercise 4.1-6 on page 67 in CLRS.

Problem 1-1. Rank the following functions by order of growth; that is, find an arrangement $g_1, g_2, \ldots, g_{16}$ of the functions satisfying $g_1 = \Omega(g_2)$, $g_2 = \Omega(g_3)$, $\ldots$, $g_{19} = \Omega(g_{16})$. Partition your list into equivalence classes such that $f(n)$ and $g(n)$ are in the same class if and only if $f(n) = \Theta(g(n))$. (The function $\log^* n$ is discussed on pages 55-56 of CLRS.)

$2^{(100^{100})}$	$(\sqrt{2})^{\lg n}$	n^2	$n!$
3^n	$\sum_{k=1}^{n} k$	$\lg^* n$	$\lg(n!)$
1	$\lg^*(\lg n)$	$\ln n$	$n^{\lg \lg n}$
$(\lg n)^{\lg n}$	2^n	$n \lg n$	$\sum_{k=1}^{n} \frac{1}{k}$

Problem 1-2. Consider the ***searching problem***: given an array $A[1 \ldots n]$ and a value v, output an index i such that $v = A[i]$ or the special value $\perp$ if v does not appear in A. If the array A is sorted, we can perform a ***binary search***: compare v with the midpoint of the array and repeat the search on one half of array, eliminating the other half from further consideration.

(a) Write pseudocode for binary search as a *recursive* procedure.

(b) Rewrite your binary search procedure in an *iterative* style.

(c) Formally state pre- and post-conditions for your iterative procedure. Write a loop invariant, and prove it correct. Then prove your entire procedure correct (what else do you need to prove?).

Problem 1-3. Solve the following recurrences. Supply both upper (O or o) and lower (Ω or ω) bounds, and make them as tight as possible. Justify your answers. (You may assume that $T(1) = 1$ in all cases, except where noted.)

(a) $T(n) = 4T(n/2) + n^3$

(b) $T(n) = T(n-2) + n^2$ (Assume $T(1) = T(2) = 1$.)

(c) $T(n) = 5T(n/3) + n\sqrt{n}$

(d) $T(n) = 2T(n/8) + \sqrt[3]{n} \ln n$ (*Warning:* the Master Theorem does not apply here.)

(e) $T(n) = 3T(n^{1/3}) + \log_3 n$ (Assume $T(3) = 1$.)

(f) $T(n) = 2T(n-1) + n^7$

Problem 1-4. Consider an $m \times n$ array A of real numbers. If we pick two rows and two columns, they intersect at exactly four elements. We say that those rows and columns satisfy the ***Monge property*** if the sum of the upper-left and lower-right elements is at most the sum of the upper-right and lower-left elements. Furthermore, we say that A is a ***Monge array*** if all choices of two rows and two columns satisfy the Monge property. Formally: A is Monge if for all i, j, k, and ℓ such that $1 \le i < k \le m$ and $1 \le j < \ell \le n$, we have:

$$A[i,j] + A[k,\ell] \le A[i,\ell] + A[k,j].$$

For example, the following array is Monge. (We have outlined two rows and columns; notice that $22 + 37 \le 29 + 44$.)

10	17	13	28	23
17	22	16	29	23
24	28	22	34	24
11	13	6	17	7
45	44	32	37	23
36	33	19	21	6
75	66	51	53	34

(a) Prove that an array is Monge if and only if adjacent rows and columns have the Monge property. More formally, prove that A is Monge if and only if for all $i = 1, 2, \ldots, m-1$ and $j = 1, 2, \ldots, n-1$:

$$A[i,j] + A[i+1,j+1] \le A[i,j+1] + A[i+1,j].$$

(*Hint:* For the "if" part, first use induction on just the rows.)

(b) Let $f_A(i)$ be the index of the column containing the leftmost minimum element of row i in matrix A. For example, we have outlined the leftmost minimum elements in the Monge array from above:

$\boxed{10}$	17	13	28	23
17	22	$\boxed{16}$	29	23
24	28	$\boxed{22}$	34	24
11	13	$\boxed{6}$	17	7
45	44	32	37	$\boxed{23}$
36	33	19	21	$\boxed{6}$
75	66	51	53	$\boxed{34}$

Prove that $f_A(1) \le f_A(2) \le \cdots \le f_A(m)$ for any $m \times n$ Monge array A.

(c) Here is a divide-and-conquer algorithm that computes $f_A(i)$ for every row i of an $m \times n$ Monge array A:

1. Construct a submatrix A' of A consisting of the even-numbered rows of A.
2. Recursively determine $f_{A'}(i)$ for each row i of A'.
3. Compute $f_A(i)$ for the rows of A, using the values returned in step 2.

Explain how to perform step 3 in $O(m+n)$ time.

(d) Write a recurrence describing the running time of the divide-and-conquer algorithm described in part **(d)**. Prove that its solution is $O(m + n \log m)$. (Assume that our representation of matrices allows step 1 to construct A' from A in $O(m)$ time.)