

Problem Set 1

MIT students: This problem set is due in lecture on **Wednesday, September 21, 2005**. The homework lab for this problem set will be held 2–4 P.M. on Sunday, September 18, 2005.

Reading: Chapters 1–4 excluding Section 4.4.

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.

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*. Convoluting and obtuse descriptions will receive low marks.

Exercise 1-1. Do Exercise 2.3-6 on page 37 in CLRS.

Exercise 1-2. Do Exercise 3.1-6 on page 50 in CLRS.

Exercise 1-3. Do Exercise 3.2-4 on page 57 in CLRS.

Exercise 1-4. Do Problem 4.3-4 on page 75 of CLRS.

Problem 1-1. Asymptotic Notation

For each of the following statements, decide whether it is **always true**, **never true**, or **sometimes true** for asymptotically nonnegative functions f and g . If it is **always true** or **never true**, explain why. If it is **sometimes true**, give one example for which it is true, and one for which it is false.

- (a) $f(n) = O(f(n)^2)$
- (b) $f(n) + g(n) = \Theta(\max(f(n), g(n)))$
- (c) $f(n) + O(f(n)) = \Theta(f(n))$
- (d) $f(n) = \Omega(g(n))$ and $f(n) = o(g(n))$ (note the little- o)
- (e) $f(n) \neq O(g(n))$ and $g(n) \neq O(f(n))$

Problem 1-2. Recurrences

Give asymptotic upper and lower bounds for $T(n)$ in each of the following recurrences. Assume that $T(n)$ is constant for $n \leq 10$. Make your bounds as tight as possible, and justify your answers.

- (a) $T(n) = 2T(n/3) + n \lg n$
- (b) $T(n) = 3T(n/5) + \lg^2 n$
- (c) $T(n) = T(n/2) + 2^n$
- (d) $T(n) = T(\sqrt{n}) + \Theta(\lg \lg n)$
- (e) $T(n) = 10T(n/3) + 17n^{1.2}$
- (f) $T(n) = 7T(n/2) + n^3$
- (g) $T(n) = T(n/2 + \sqrt{n}) + \sqrt{6046}$
- (h) $T(n) = T(n - 2) + \lg n$
- (i) $T(n) = T(n/5) + T(4n/5) + \Theta(n)$
- (j) $T(n) = \sqrt{n} T(\sqrt{n}) + 100n$

Problem 1-3. Unimodal Search

An array $A[1..n]$ is **unimodal** if it consists of an increasing sequence followed by a decreasing sequence, or more precisely, if there is an index $m \in \{1, 2, \dots, n\}$ such that

- $A[i] < A[i + 1]$ for all $1 \leq i < m$, and
- $A[i] > A[i + 1]$ for all $m \leq i < n$.

In particular, $A[m]$ is the maximum element, and it is the unique “locally maximum” element surrounded by smaller elements ($A[m - 1]$ and $A[m + 1]$).

- (a) Give an algorithm to compute the maximum element of a unimodal input array $A[1..n]$ in $O(\lg n)$ time. Prove the correctness of your algorithm, and prove the bound on its running time.

A polygon is **convex** if all of its internal angles are less than 180° (and none of the edges cross each other). Figure 1 shows an example. We represent a convex polygon as an array $V[1..n]$ where each element of the array represents a vertex of the polygon in the form of a coordinate pair (x, y) . We are told that $V[1]$ is the vertex with the minimum x coordinate and that the vertices $V[1..n]$ are ordered counterclockwise, as in the figure. You may also assume that the x coordinates of the vertices are all distinct, as are the y coordinates of the vertices.

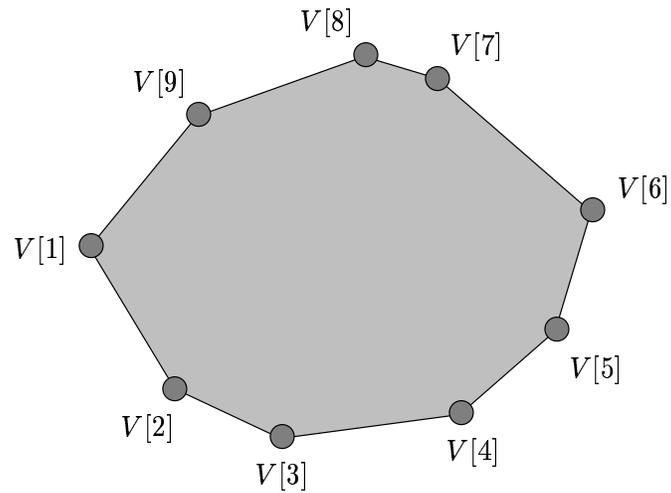


Figure 1: An example of a convex polygon represented by the array $V[1..9]$. $V[1]$ is the vertex with the minimum x -coordinate, and $V[1..9]$ are ordered counterclockwise.

- (b) Give an algorithm to find the vertex with the maximum x coordinate in $O(\lg n)$ time.
- (c) Give an algorithm to find the vertex with the maximum y coordinate in $O(\lg n)$ time.