

Homework 1

Assigned: September 9

Due: September 21

Problem 4 has been changed

Please respect the following guidelines for writing pseudocode:

1. C instructions are fine. But do not write object-oriented additions. Do not declare or use any class. Declare only procedures (if necessary) and explain in words what each procedure does, and what is the use of each parameter.
2. One instruction per line
3. Match the brackets with a horizontal line
4. Number your lines
5. Write down if your array is indexed $0 \dots n - 1$ or $1 \dots n$ (Problem 4 forces you to use $1 \dots n$).

Problem 1 Order the following list of functions by the big-Oh notation. Group together (for example, by underlining) those functions that are big-Theta of one another.

$6n \log n$	2^{100}	$\log \log n$	$\log^2 n$	$2^{\log n}$
2^{2^n}	$\lceil \sqrt{n} \rceil$	$n^{0.01}$	$1/n$	$4n^{3/2}$
$3n^{0.5}$	$5n$	$\lfloor 2n \log^2 n \rfloor$	2^n	$n \log_4 n$
4^n	n^3	$n^2 \log n$	$4^{\log n}$	$\sqrt{\log n}$

Hint: When in doubt about two functions $f(n)$ and $g(n)$, consider $\log f(n)$ and $\log g(n)$ or $2^{f(n)}$ and $2^{g(n)}$.

Problem 2 Describe a method for finding both the minimum and maximum of n elements using fewer than $3n/2$ comparisons between elements.

Hint: First construct a group of candidate minimums and a group of candidate maximums.

Note that only the number of comparisons between elements matter, not the running time. Present pseudocode, count the number of comparisons between elements done in the worst case, and argue your solution is correct.

Problem 3 An n -degree *polynomial* $p(x)$ is an equation of the form

$$p(x) = \sum_{i=0}^n a_i x^i,$$

where x is a real number and each a_i is a constant.

- a. Describe a simple $O(n^2)$ time method for computing $p(x)$ for a particular value of x , given as input x and the length- $(n + 1)$ array of coefficients A .

b. Consider now a rewriting of $p(x)$ as

$$p(x) = a_0 + x(a_1 + x(a_2 + x(a_3 + \dots + x(a_{n-1} + xa_n)\dots))),$$

which is known as *Horner's method*. Using the big-Oh notation, characterize the number of multiplications and additions this method of evaluation uses.

Present pseudocode.

Problem 4 Give pseudocode for the following problem: given an array $A[1 \dots n]$ containing the numbers $1, 2, \dots, n$ and representing a permutation, modify A to represent the next lexicographic permutation.

Analyse running time (which should not exceed $O(n)$) and argue correctness (why the output is lexicographically bigger than the input? why other permutations lexicographically bigger than the input are bigger than the output?).

Problem 5 1. Draw the 11-item hash table resulting from hashing the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, and 5, using the hash function $h(i) = (2i + 5) \bmod 11$ and assuming collisions are handled by chaining.

2. What is the result of the previous exercise, assuming collisions are handled by linear probing?
3. Show the result of Exercise 1 above in this problem, assuming collisions are handled by quadratic probing, up to the point where the method fails because no empty slot is found.
4. What is the result of Exercise 1 above in this problem, assuming collisions are handled by double hashing using a secondary hash function $h'(k) = 7 - (k \bmod 7)$?