Please solve the following exercises and submit **BEFORE 12:00 pm (noon) of Thursday 27th, November.**

Solve the following two exercises from **assignment 8** in case you have **NOT** already solved them.

Assignment 8 Exercises:

Exercise 9 (10 points)

Show that if there were a coin worth 12 cents, the greedy algorithm described in class using quarters, 12-cent coins, dimes, nickels, and pennies would not always produce change using the fewest coins possible.

Exercise 10 (10 points)

Show that a greedy algorithm that schedules a set of talks in a lecture hall by selecting at each step the talk that overlaps the fewest with other talks does not always produce an optimal schedule.

Assignment 9 Exercises:

Exercise 1 (10 points)

Determine whether each of these functions is $O(x^2)$, $\Omega(x^2)$ and $\theta(x^2)$.

a) $f(x)= 17x+11$

b) $f(x)= x^2+1000$

c) $f(x)= x \log x$

d) $f(x)= x^4/2$

e) $f(x)= 2^x$

f) $f(x)= \lfloor x \rfloor \cdot \lceil x \rceil$

Exercise 2 (10 points)

Arrange the functions $\sqrt{n}$, 1000 log n, n log n, 2n!, 2^n, 3^n, and $n^2/1,000,000$ in a list so that each function is a big-*O* of the next function.

Exercise 3 (10 points)

Give a big-O estimate for each of these functions. For the function *g* in your estimate f(x) is O(g(x)), use a simple function *g* of smallest order.

a) $(n^3 + n^2 \log n)(\log n + 1) + (17 \log n + 19)(n^3 + 2)$

b) $(2^n + n^2)(n^3 + 3^n)$

c) $(n^n + n2^n + 5^n)(n! + 5^n)$

Exercise 4 (10 points)

Show that each of these pairs of functions are of the same order.

a) $2x^2+x-7,\ x^2$

b) $\lfloor x+1/2 \rfloor,\ x$

c) $\log(x^2 + 1),\ \log x^2$

d) $\log_{10} x\ ,\ \log_2 x$

Exercise 5 (10 points)

Given a real number x and a positive integer k, determine the number of multiplications used to find x^{2^k} starting with x and successively squaring (to find x^2, x^4, and so on). Is this a more efficient way to find x^{2^k} than by multiplying x by itself the appropriate number of times?

Exercise 6 (10 points)

What is the largest n for which one can solve a certain problem with input size n within a day using an algorithm that requires f(n) bit operations, where each bit operation is carried out in 10^{-11} seconds, with these functions f (n)?

a) $\log n$ **b)** $1000n$ **c)** n^2

d) $1000n^2$ **e)** n^3 **f)** 2^n

g) 2^{2n} **h)** 2^{2^n}

Exercise 7 (10 points)

An algorithm is called **optimal** for the solution of a problem with respect to a specified operation if there is no algorithm for solving this problem using fewer operations.

a) Is the linear search algorithm optimal with respect to the number of comparisons of integers (not including comparisons used for bookkeeping in the loop)?

b) Show that the following Algorithm A is an optimal algorithm with respect to the number of comparisons of integers. [*Note:* Comparisons used for bookkeeping in the loop are not of concern here.]

ALGORITHM A: Finding the Maximum Element in a Finite Sequence.

procedure *max*($a_1, a_2, \ldots, a_n$: integers)

max := a_1

for i := 2 **to** n

  if *max*<a_i **then** *max*:=a_i

return *max*{*max* is the largest element}

Exercise 8 (10 points)

Describe an algorithm that takes as input a list of n integers in nondecreasing order and produces the list of all values that occur more than once. (Recall that a list of integers is nondecreasing if each integer in the list is at least as large as the previous integer in the list). Determine the worst-case time complexity in terms of comparisons for your algorithm.

Exercise 9 (10 points)

Find the complexity of a brute-force algorithm for scheduling the talks by examining all possible subsets of the talks. [*Hint:* Use the fact that a set with n elements has 2^n subsets.]

Exercise 10 (10 points)

Describe how the number of comparisons used in the worst case changes when these algorithms are used to search for an element of a list when the size of the list doubles from n to 2n, where n is a positive integer.

a) linear search **b)** binary search