Due 23:59 Jan 28 (Sunday). There are 100 points in this assignment. Submit your answers (**must be typed**) in pdf file to CourSys
`https://coursys.sfu.ca/2024sp-cmpt-307-d1/`.
Submissions received after 23:59 will get penalty of reducing points: 20 and 50 points deductions for submissions received at $[00:00, 00:10]$ and $(00:10, 00:30]$ of Jan 29, respectively; no points will be given to submissions after 00 : 30 of Jan 29.

In the assignment, $\log n = \log_2 n$.

1. 10 points (Ex. 1.2-2, 1.2-3 of text book)

   (a) For inputs of size n, assume insertion sort runs in $8n^2$ steps and merge sort runs in $64n \log n$ steps. For which value n does insertion sort run faster than merge sort on a same machine?

   (b) What is the smallest value of n such that an algorithm of running time $100n^2$ runs faster than an algorithm of running time 2^n on a same machine?

2. 10 points

   Suppose you have algorithms with the six running times listed below. (Assume these are the exact number of operations performed as a function of the input size n.) Suppose you have a computer that can perform 10^{10} operations per second, and you need to compute a result in at most an hour of computation. For each of the algorithms, what is the largest input size n for which you would be able to get the result within an hour?

   (1) n^2, (2) n^3, (3) $100n^2$, (4) $n \log n$, (5) 2^n, (6) 2^{2^n}.

3. 20 points (Ex 2.1-4 of text book)

   Consider the searching problem:

   Input: A sequence of n numbers $A = (a_1, a_2, .., a_n)$ and a value v.
   Output: An index i such that $v = A[i]$ or the special value nil if v is not in A.

   Write pseudocode for linear search which scans through the sequence, looking for v. Using a loop invariant, prove your algorithm is correct. Make sure your loop invariant fulfills the three properties (initialization, maintenance, termination).

4. 20 points (Ex 2.3-6 of text book)

   Referring back to the searching problem (Ex 2.1-4), assume the sequence A is sorted, we can check the midpoint of A and eliminate half of the elements of A from further consideration. The binary search algorithm repeats this procedure, halving the size of the remaining portion of A each time. Write pseudocode, either iterative or recursive, for binary search. Argue that the worst-case running time of binary search is $\Theta(\log n)$.

5. 10 points (Problem 3-2 of text book)

   Relative asymptotic growths: Indicate, for each pair of expressions $f(n)$ and $g(n)$ in the table below, whether $f(n)$ is O, o, Ω, ω, or Θ of $g(n)$. Assume that $k \geq 1$, $\epsilon > 0$,

and $c > 1$ are constants. Your answer should be in the form of the table with "yes" or "no" written in each box. The answers for $f(n) = \log^k n$ and $g(n) = n^\epsilon$ are given in the 1st row of table as an example.

$f(n)$	$g(n)$	O	o	Ω	ω	Θ
$\log^k n$	n^ϵ	yes	yes	no	no	no
n^k	c^n					
2^n	$2^{n/2}$					
$n^{\log c}$	$c^{\log n}$					
$\log(n!)$	$\log(n^n)$					

6. 10 points

   Implement the conventional matrix multiplication and the fast matrix multiplication by Strassen. For $n = 2^i$, $i = 4, 5, 6, 7, 8, 9, 10$, report the running times of your programs for computing the product of two $n \times n$ matrices by the two methods. You can use any programming language, but the same computing platform for both methods. You only need to report the running times, no need to submit the programs.

7. 10 points

   How would you modify Strassen's algorithm to multiply $n \times n$ matrices in which n is not an exact power of 2? Show that the resulting algorithm runs in time $O(n^{\log 7})$.

8. 10 points (Ex 4.5-1 of text book)

   Use the master method to give tight asymptotic bounds for the following recurrences. (a) $T(n) = 2T(n/4) + 1$. (b) $T(n) = 2T(n/4) + \sqrt{n}$. (c) $T(n) = 2T(n/4) + n$. (d) $T(n) = 2T(n/4) + n^2$.