# Homework Assignment #2

## Note

This assignment is due 2:10PM Monday, October 8, 2018. Please write or type your answers on A4 (or similar size) paper. Drop your homework by the due time in the Algorithms box placed in the IM office on the 7-th floor of Management College Building 1. Late submission will be penalized by 20% for each working day overdue. You may discuss the problems with others, but copying answers is strictly forbidden.

## Problems

There are five problems in this assignment, each accounting for 20 points. (Note: problems marked with "(X.XX)" are taken from [Manber 1989] with probable adaptation.)

1. (3.5) For each of the following pairs of functions, say whether $f(n) = O(g(n))$ and/or $f(n) = \Omega(g(n))$. Justify your answers.

| | $f(n)$ | $g(n)$ |
|---|---|---|
| (a) | $100n + \log n$ | $n + (\log n)^2$ |
| (b) | $n2^n$ | $3^n$ |

2. (3.13) Use Equation 1, to prove that, for every positive integer $k$,

   $\sum_{i=1}^{n} i^k = O(n^{k+1})$.

   **Bounding a summation by an integral**
   If $f(x)$ is a monotonically increasing continuous function, then

$$\sum_{i=1}^{n} f(i) \leq \int_{x=1}^{x=n+1} f(x)dx. \tag{1}$$

3. (3.26) Find the asymptotic behavior of the function $T(n)$ defined by the recurrence relation

   $T(n) = T(n/2) + \sqrt{n}$, $T(1) = 1$.

   You can consider only values of $n$ that are powers of 2.

4. What is the time complexity of the following program segment? Your answer should be as tight as possible.

```
for i from 1 to n
  for j from 1 to i
    for k from 1 to n
      ··· (constant-time operations)
    end for
  end for
end for
```

5. Consider the following procedure of insertion sort. In order to sort a list of $n$ elements, we remove the maximum element from the list, recursively sort the rest $n-1$ elements, and then append the maximum element to the sorted $n-1$ elements. Assume (1) the running time of removing the maximum element in a list of $n$ elements is $n$, and (2) the running time of appending one element to a list is 1. Write a recurrence relation for the running time of this version of insertion sort. Give an exact solution for the recurrence relation.