CIS 606 **Homework 1** **Fall 2020**

Name: **Prudhvi Reddy Araga**

Login ID: **2788245**

1. We can sum $\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$ inside text or use the displayed math like below:

$$\sum_{k=1}^{n} k = 1 + 2 + \cdots + n = \frac{n(n+1)}{2}$$

2. The recurrence (1) shows the worst-case running time $T(n)$ of mergesort:

$$T(n) = \begin{cases} c & \text{if } n = 1, \\ 2T(n/2) + cn & \text{if } n > 1. \end{cases} \tag{1}$$

Using the master theorem in Chapter 4, we can get $T(n) = \Theta(n \log n)$.

3. The recurrence (2) shows the worst-case running time $T(n)$ of binary search:

$$T(n) = \begin{cases} c & \text{if } n = 1, \\ T(n/2) + c & \text{if } n > 1. \end{cases} \tag{2}$$

Using the master theorem in Chapter 4, we can get $T(n) = \Theta(\log n)$.

4. Browse `https://www.cs.dartmouth.edu/~thc/clrscode/clrscode3e.pdf` to learn how to use the clrscode3e package in LaTex to typeset pseudocode.

```
INSERTION-SORT(A)
for j = 2 to A.length
    key = A[j]
    // Insert A[j] into the sorted sequence A[1 .. j - 1].
    i = j - 1
    while i > 0 and A[i] > key
        A[i + 1] = A[i]
        i = i - 1
    A[i + 1] = key
```

5. If $f(n) = O(n^{log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{log_b a})$.