

Solutions to Mid-Term Exam #1

Part I: Algorithm Efficiency Using Big-O Notations. (16 points)

1. True
2. Since $T(n) = O(n^2)$, we have
$$\begin{aligned}T(n) &\leq C \cdot n^2 && \text{(where } C \text{ is a constant)} \\&\leq C \cdot n^3 && \text{(where } n \geq 1\text{)} \\&= O(n^3)\end{aligned}$$
3. $T(n) = \sum_{i=1}^n (i + 1) = O(n^2)$
4. $T(0) = 1$
 $T(n) = T(n/2) + 1$
By telescoping principle (or backward substitution), we have
 $T(n) = O(\log n)$

Part II: Programming (10 points)

```
// insert x if it does not exist
void SortedList::insert(int x) {
    Node* prev = NULL;
    Node* cur = first;
    while (cur && cur->data > x) {
        prev = cur;
        cur = cur->next;
    }
    if (cur && cur->data == x) //x exists, return
        return;
    Node* p = new Node(x, cur);
    if (!p)
        Return;
    if (!prev)
        first = p;
    else
        prev->next = p;
}
```

```
void SortedList::remove(int x) {
    Node* prev = NULL;
    Node* cur = first;
    while (cur && cur->data > x) {
        prev = cur;
        cur = cur->next;
    }
    if (cur && cur->data == x) { //found
        if (!prev)
            first = cur->next;
        else
            prev->next = cur->next;
        delete cur;
    }
}
```