

# Assignment # 1

1. (Weight: 30%) Calculate  $T(n)$  and  $O(n)$  for the following algorithms (Quick Sort) for the average and worst case:

```
void quick_sort(int first, int last, std::vector<int>& arr) {
    if (last - first > 1) {
        // There is data to be sorted.
        // Partition the table.
        int pivot = partition(first, last, arr);
        // Sort the left half.
        quick_sort(first, pivot, arr);
        // Sort the right half.
        quick_sort(pivot + 1, last, arr);
    }
}

int partition(int first, int last, std::vector<int>& arr) {
    .
    int up = first + 1;
    int down = last - 1;
    do {
        while ((up != last - 1) && arr[first] >= arr[up]) {
            ++up;
        }
        while (arr[first] < arr[down]) {
            --down;
        }
        if (up < down) {
            // if up is to the left of down,
            swap(arr[up], arr[down]);
        }
    } while (up < down); // Repeat while up is left of down.
    swap(arr[first], arr[down]);
    return down;
}
```

2. (Weight: 30%) Use substitution, summation, or recursion tree method to solve the following recurrence relations.

(a)  $T(n) = 2T(n/2) + n \lg n$

$$T(1) = \Theta(1)$$

(b)  $T(n) = 2T(n-1) + 5^n$

$$T(0) = 8$$

**3.** (Weight: 40%) Use master method to determine and use theta ( $\Theta$ ) notation to represent asymptotic growth rate for each  $T(n)$  of the following recurrence relations. Assume  $T(n)$  is constant for  $n \leq 4$ . You need to show clear steps to justify your answers.

**(a)**  $T(n) = 9T(n/2) + n^3 \lg n$

**(b)**  $T(n) = 9T(n/3) + n^2$

**(c)**  $T(n) = 6T(n/2) + n^3$