1. (4 points) Exercise 2.2-2

2. (2 points) Use mathematical induction to show that when n is an exact power of 3, the solution of the recurrence

\[ T(n) = \begin{cases} 
9 & \text{if } n = 3 \\
6T(n/3) + \frac{1}{3}n^2 & \text{if } n = 3^k, \text{ for } k > 1
\end{cases} \]

is \( T(n) = n^2 \)

3. (4 points) Problem 2-1: Insertion sort on small arrays in merge sort

4. (3 points) Consider the following program and recursive function.

```c
void main() {
    int A[3] = {1, 2, 3};
    Z(A, A.length, 0);
}
void Z(int A[], int n, int k) {
    if (k == n-1) {
        for (int i=0; i<n; i++) cout << A[i] << " ";
        cout << endl;
    }
    else {
        for (int i=k; i<n; i++) {
            swap(A[i], A[k]);
            Z(A, n, k+1);
            swap(A[i], A[k]);
        }
    }
}
```

4a. Demonstrate the execution, show the output, and explain what the program accomplishes.

4b. Give a recurrence equation describing the worst-case behavior of the program.

4c. Solve the recurrence equation.

5. (6 points) Give big-O bounds for \( T(n) \) in each of the following recurrences. Use induction, iteration or Master Theorem. You may assume \( T(1)=1 \) in all cases.

5a. \( T(n) = T(n-1) + n^2 \)

5b. \( T(n) = 5T(n/3) + n*n^{1/2} \)

5c. \( T(n) = T(n/4) + T(n/2) + n^2 \)

6. (6 points) Problem 4-2: Parameter-passing costs