

1. Given the following function that evaluates a polynomial whose coefficients are stored in an array:

```
double evaluate(double[] coefficients, int n, double x)
    double result = coefficients[0];
    double power = 1;
    for (int i = 1; i < n; i++)
        power = power * x;
        result = result + coefficients[i] * power;
    return result;
```

Let  $n$  be the length of the array. Determine the number of additions and multiplications that are performed in the worst case as a function of  $n$ .

2. Suppose the number of steps required in the worst case for two algorithms are as follows:
  - Algorithm 1:  $f(n) = 3n^2 + 5$
  - Algorithm 2:  $g(n) = 53n + 9$

Determine at what point algorithm 2 becomes more efficient than algorithm 1.

Consider the following iterative function for problems 3 and 4.

```
int triangular(int n)
{
    int result = 0;
    for (int i = 1; i <= n; i++)
        result += i;
    return result;
}
```

3. Rewrite the function `triangular` using recursion and add preconditions and postconditions as comments.
4. Prove by induction that the recursive function you wrote in the previous problem is correct.