1. In insertion sort, we follow such a procedure: we first find the smallest element in array $A$ and exchange it with $A[1]$, then we find the second smallest element and exchange it with $A[2]$. We repeat the procedure $(n - 1)$ times where $n$ is the number of elements in $A$. Write pseudocode for this algorithm. Analyze its worse case complexity.

2. To compute a polynomial $P(x) = \sum_{k=0}^{n} a_k x^k$, we can use the following method:

$$P(x) = a_0 + x(a_1 + x(a_2 + ... + x(a_{n-1} + xa_n)))$$

The algorithm’s pseudocode is as follows:

```plaintext
y = 0
i = n
while i \geq 0 do
    y = a_i + x \cdot y
    i = i - 1
end while
```

(a) Show that this code is correct.

(b) What’s the asymptotic running time of this code, assuming that your machine can do addition and multiplication in a single step?

(c) How does it compare with directly evaluating the polynomial?

3. Show that for any constant $a$ and $b$, where $b > 0$

$$(n + a)^b = \Theta(n^b)$$

4. Is $2^{n+1} = O(2^n)$? Is $2^{2n} = O(2^n)$? Justify your answer.

5. Is a $O(n^2)$ algorithm always slower than a $O(n)$ algorithm? Why?

6. Show the relative asymptotic growth of the following pairs ($k > 1, e > 0, c > 1$) and justify your result:

(a) $\log^k(n)$ and $n^e$
(b) $n^k$ and $c^n$
(c) $\sqrt{n}$ and $n^{\sin(n)}$
(d) $2^n$ and $2^{n/2}$
(e) $n^{\log(c)}$ and $c^{\log(n)}$

7. Calculate $2^{125} \mod 127$. (Hint 127 is a prime).

8. Use the recursion tree method to show that $T(n)$ is $O(n \log(n))$ if $T(n) = 2T(n/2 + 17) + n$. 

1