EECS 3101: Design and Analysis of Algorithms
Tutorial 3

1. Prove that the following algorithm that computes the n^{th} Fibonacci number is correct.

```
FIB(n)
1  if n = 0
2     then return 0
3     else  last ← 0
4           current ← 1
5           for j ← 2 to n
6           do temp ← last + current
7              last ← current
8              current ← temp
9  return current
```

2. Prove that the following algorithm for exponentiation is correct.

```
POWER(y, z)
1  // return y^z where y ∈ R, z ∈ N
2  x ← 1
3  while z > 0
4  do if odd(z)
5        then x ← x * y
6     z ← ⌊z/2⌋
7     y ← y^2
8  return x
```

3. For the following segment of code, find the value returned by $T(n)$.

```
T(n)
1  if n == 1
2     then return 1
3     else  sum = 0;
4           for i = 1 to n − 1
5           do sum = sum + T(i) + T(n − i)
6           return (sum + 2n)
```

4. Suppose $T(n)$ is a constant for $n \leq 2$. Solve for $T(n)$ asymptotically and justify your answer.

$$T(n) = 7T(n/3) + n^2.$$

5. Solve the following recurrence assuming that $T(n)$ is constant for sufficiently small n. Justify your answer.

$$T(n) = T(n-2) + 2\lg n.$$