HOMEWORK-3
Total Points: 65

1. [10 Points] Prove that 2^{20n} is not $O(2^{n})$.

2. [5 Points] Prove that 2^{20+n} is $O(2^{n})$.

3. [10 Points] What is the time complexity of the following algorithm? Justify your answer.

```
long long foo(long long n) {
  long long i = 0;
  while (i < n) {
     if (i % 5 == 0) {
       i = i + 6;
     }
     else if (i % 5 == 1) {
       i = i + 1;
     }
     else if (i % 5 == 2) {
       i = i + 16;
     }
     else if (i % 5 == 3) {
       i = i + 11;
     }
     else if (i % 5 == 4) {
       i = i + 21;
     }
  }
  return i;
}
```

4. [15 Points] What is the time complexity of the following algorithm? Justify your answer.

```
long long foo(long long n) {
  long long i = 0;
  while (i < n) {
     if (i % 5 == 0) {
       i = i +  6;
     }
     else if (i % 5 == 1) {
       i = i + 1;
     }
     else if (i % 5 == 2) {
       i = i + 16;
     }
     else if (i % 5 == 3) {
       i = i + 11;
     }
     else if (i % 5 == 4) {
       i = i * 5;
     }
  }
  return i;
}
```

5. [15 Points] Compute the lower bound and upper bound on the number of operations performed by the recursive algorithm for Fibonacci. Use the expansion method discussed in class.
   The recurrence relation for the lower bound is:
   T(0) = T(1) = 2
   T(n) >= 2T(n-2) + c
   The recurrence relation for the upper bound is:
   T(0) = T(1) = 2
   T(n) <= 2T(n-1) + c

6. [10 Points] Write a recursive algorithm to check whether a number “n” is prime. The time-complexity of your algorithm should be less than O(n). Discuss why your algorithm is correct. What is the time complexity of your algorithm? Justify your answer.