# SCC 120: Week 13 Workshop Problems

**Question 1.** The Fibionacci numbers for $n = 0, 1, \ldots$ are defined as follows.

- $\mathit{fib}(0) = 0$;
- $\mathit{fib}(1) = 1$;
- $\mathit{fib}(n) = \mathit{fib}(n-1) + \mathit{fib}(n-2)$;

Consider the following piece of code that calculates the Fibionacci numbers.

```
int fib(int n) {
    if (n is 0)
        return 0
    else if (n is 1)
        return 1
    else
       return fib(n-1) + fib(n-2)
}
```

What is the worst case complexity of the above code?

**Question 2.** Write a program to calculate the Fibionacci numbers that runs in linear time in the worst case.

**Question 3.** Let $p$ and $q$ be propositional variables and $\wedge$ and $\neg$ the usual logical operators. List all formulas entailed by $p$ (that is, if $p$ were true, what other formulas would be true?).

**Question 4.** What is an approach for determining if a formula is unsatisfiable, that is, inconsistent? What is its lower bound on its worst case complexity? Is this better or worse than satisfiability or validity of a formula?

**Question 5.** Consider a variant of the subset sum problem called *min-subset-sum*. For min-subset-sum, this is what you must do.

- If there is a subset that sums to 0, you must return a smallest such subset
- Else return *null*.

1. Is min-subset-sum a decision problem?
2. For the same set can you get different answers?
3. Describe an approach for solving min-subset-sum?
4. Do you think it would be correct to claim that your approach is $\Omega(n)$? How about $\Omega(2^n)$?

**Question 6.** Use the undecidability of the *Halting* problem to show the undecidability of the *Dead Code* problem.

**Question 7.** Indicate the true statements

1. NP-Completeness problems are in a sense the hardest problems in NP
2. If propositional satisfiability (SAT) has a polynomial time solution, then so does Hamiltonian Cycle (HC)
3. If SAT has a polynomial time solution, then so does linear search
4. If P=NP, then solving any problem is as easy as verifying it
5. To show HC is NP-Complete, in addition to showing it is in NP, I need to give a reduction from HC to SAT
6. If a problem is NP-Completeness, that is proof of the intractability of the problem