COL351: Analysis and Design of Algorithms
Instructor: Ragesh Jaiswal

1. This is a recap. of a few proof techniques that you studied in the Discrete Mathematics course. We will use the following definition of even and odd numbers in the example problems that follow:

   Odd/even numbers: An integer n is called even iff there exists an integer k such that $n = 2k$. An integer n is called odd iff there exists an integer k such that $n = 2k + 1$.

   - Direct proof: Used for showing statements of the form p implies q. We assume that p is true and use axioms, definitions, and previously proven theorems, together with rules of inference, to show that q must also be true.
     - Give a direct proof of the statement: "If n is an odd, then n^2 is odd".
   - Proof by contraposition: Used for proving statements of the form p implies q. We take $\neg q$ as a premise, and using axioms, definitions, and previously proven theorems, together with rules of inference, we show that $\neg p$ must follow.
     - Prove by contraposition that "if n^2 is odd, then n is odd".
   - Proof by contradiction: Suppose we want to prove that a statement p is true and suppose we can find a contradiction q such that $\neg p$ implies q. Since q is false, but $\neg p$ implies q, we can conclude that $\neg p$ is false, which means that p is true. The contradiction q is usually of the form $r \wedge \neg r$ for some proposition r.
     - Give a proof by contradiction of the statement: "at least four of any 22 days must fall on the same day of the week"
   - Counterexample: Suppose we want to show that the statement for all x, $P(x)$ is false. Then we only need to find a counterexample, that is, an example x for which $P(x)$ is false.
     - Show that the statement "Every positive integer is the sum of squares of two integers" is false.
   - Mathematical Induction: This was discussed in the lecture.
     - Show using induction that for all positive integer n, $1+2+3+...+n = n \cdot (n+1)/2$.
     - Show using induction that for all positive integers n, $1 + 2^1 + 2^2 + ... + 2^n = 2^{n+1} - 1$.

2. Assume you have functions f and g such that $f(n)$ is $O(g(n))$. For each of the following statements, decide whether it is true or false and give a proof or counterexample.

- $\log_2 f(n)$ is $O(\log_2(g(n)))$
- $2^{f(n)}$ is $O(2^{g(n)})$
- $f(n^2)$ is $O(g(n^2))$

3. The Fibonacci numbers $F_0, F_1, F_2, ...$ are defined by the rule

$$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}$$

Show by induction that $F_n = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n \right]$.

4. Discuss the running time of the following algorithm:

```
Fib(n)
  - If (n = 0 or n = 1) then return(n)
  - return(Fib(n − 1) + Fib(n − 2))
```

Following is the recurrence relation for the running time of the above recursive algorithm:

$$T(n) \leq T(n-1) + T(n-2) + dn; \quad T(0) \leq d; \quad T(1) \leq d,$$

where d is some constant. One way to solve and get an upper bound for this recurrence relation is using substitution method. Here, we make a guess on the bound and then prove the the bound is correct using induction. Let us make the following guess: $T(n) \leq cn2^n$ for all $n \geq 2$. We will show that for a suitable choice of constant c, $T(n) \leq cn2^n$ for all $n \geq 2$. Let us try to prove the above statement using induction. Consider $n = 2$ for the basis step. We have $T(2) \leq T(1) + T(0) + 2d = 4d$. So, as long as $c \geq d/2$, we have that $T(2) \leq c \cdot 2 \cdot 2^2$. For the inductive step, assume that $T(i) \leq ci$ for $i = 2, 3, 4, ..., k-1$. We will show that $T(k) \leq c(k)2^k$. We show this using the recurrence relation:

$$\begin{aligned} T(k) &\leq T(k-1) + T(k-2) + dk \\ &\leq c(k-1)2^{k-1} + c(k-2)2^{k-2} + dk \\ &\leq c2^{k-2}(2k-2+k-2) + dk \\ &\leq (3/4) \cdot ck2^k + dk \\ &\leq ck2^k \end{aligned}$$

The last inequality is true as long as $dk \leq ck2^k/4$. So, if we choose $c = d/2$, then both the basis step and the inductive step go through. So, we get that $T(n) = O(n2^n)$.