In lab exercises

1. Trace the execution of the SCC Algorithm on the graph given in class and show that it correctly outputs the correct set of SCC labels

2. Finish the proof that the SCC Algorithm is correct.

3. (2.4) Rank the following list of function in increasing order of growth. Sometimes it may help to take the logarithm of two functions and compare their logs. See also the first homework problem.

   - $g_1(n) = 2^{\sqrt{\lg n}}$
   - $g_2(n) = 2^n$
   - $g_3(n) = n^{4/3}$
   - $g_4(n) = n \lg^3 n$
   - $g_5(n) = n^{\lg n}$
   - $g_6(n) = 2^{2^n}$
   - $g_7(n) = 2^{n^2}$
   - $g_8(n) = 2^{n+1}$
   - $g_9(n) = 2^{2n}$
   - $g_{10}(n) = n \lg n$
   - $g_{11}(n) = \lg \lg n$

4. (3.6) Let $G$ be an undirected connected graph and let $u$ be a vertex in $G$. If we run $\mathrm{D}FS(G)$ starting from $u$, we will get a single tree $T$ rooted at $u$ containing all vertices in $G$. Suppose a BFS starting from $u$ yields the same tree $T$. Show $G = T$.

5. (3.11) Let $C_1, C_2, \ldots C_n$ be a set of networked computers. Each time two computers $C_i$ and $C_j$ communicate, a log message $(C_i, C_j, t_k)$ is recorded to indicate that $C_i$ and $C_j$ exchanged information at time $t_k$. Suppose machine $C_a$ is infected with a virus at time $t_a$. Could the virus have spread to $C_b$ by time $t_b$? Assume that we are given $m$ log messages of the form $(C_i, C_j, t_k)$ and that a machine $C_b$ could receive a virus from another machine $C_a$, if there exists a path of communications from $C_a$ to the eventual target $C_b$, such that the time stamps along the path are monotonically non-decreasing (not moving backward in time). Given the machines, the log messages, $C_a, C_b, t_a$, and $t_b$, design an algorithm to determine if $C_b$ could be infected by a virus injected into $C_a$ at time $t_a$ by the time $t_b > t_a$. What is the run time of your algorithm.

6. (2.8, the iPhone FreeFall app) A small start up company is releasing a highly anticipated smartphone in the near future. The primary features of the new phone are a supposedly rugged design and a fancy proprietary charging and data cable. The determine the ruggedness of the phone, beta testers are trying to determine the maximum height from which they can drop the phone and have it not break. They conveniently have a building with $n$ floors and they wish to determine the highest floor from which they can safely drop the phone. Given $\log n$ phones, one could use binary search to identify the highest safe floor, but the phones are in short supply as each phone requires the tears of a unicorn to manufacture. Suppose you are given a small number, $k$, of phones. Design an algorithm that can determine the maximum safe floor in the fewest number of drops, $f_k(n)$. Try to find solutions for $k = 1, 2$, and 3.