1. (25 points) Graph search and topological sorting
   a. (10 points) Consider a breadth-first search of the following directed graph, starting at vertex 0. Draw the breadth-first tree and the time stamps in which the vertices are visited. Iterate through the neighbors of v in increasing order. Did BFS eventually visit all vertices in the graph?

   ![Graph for BFS](image)

   b. (10 points) Consider a depth-first search of the following directed graph, starting at vertex 0. Draw the depth-first tree, the time stamps in which the vertices are first visited, and the time stamps in which the vertices are finished. (Iterate through the neighbors of v in increasing order.) Did DFS eventually visit all vertices in the graph?

   ![Graph for DFS](image)

c. (5 points) Show the topologically sorted order of the vertices of the above graph, using the results you obtained above in 1(b).
2. (15 points) **P and NP**
   Determine whether the following statement is true or false. Briefly justify your answers.
   
a. **P** is the class of all problems that can be solved in polynomial time.
   
b. **NP** is the class of all decision problems that cannot be solved in polynomial time.
   
c. A problem is **NP-complete** if it is both **NP-hard** and in **NP**.
   
d. If SAT can be solved in polynomial time, all problems in **NP** can be solved in polynomial time.
   
e. Suppose $\Pi_1$ and $\Pi_2$ are two decision problems and $\Pi_1$ is polynomial time reducible to $\Pi_2$, i.e., $\Pi_1 \leq_p \Pi_2$. If $\Pi_1$ is in **NP**, then $\Pi_2$ is also in **NP**.

3. (Extra credit: 15 points) **Suffix Tree**
   
a. Draw a suffix tree for the string `taataataaa`. Label the edges and terminal nodes explicitly.
b. **Shortest nonrepeated substring.** Design an efficient algorithm for finding the shortest substring that appears in a text only once.

c. **Shortest signature substrings.** Design an efficient algorithm to find the minimum $l$ for a set of strings $T_1, T_2, \ldots, T_k$, such that there exist a unique “signature” substring of length $l$ for each string. For example, if $T_1 = \text{ACGACGTA}$, $T_2 = \text{ACTATGAC}$, and $T_3 = \text{GATAGTA}$, the smallest $l = 2$, since a signature of length 2 can be found for each string: CG only appears in $T_1$, CT only in $T_2$ and AG only in $T_3$.

4. (Extra credit - 10 points) Please provide any comments/suggestions about the lecture, recitation, and homework. Use additional page if necessary.