Directions:

- Write your solution using \LaTeX. Submit this homework as file `hw3.tex` using github.
- Try to make your answers as clear and concise as possible.
- This is an individual homework. It’s ok to discuss approaches and strategies at a high level. However, you should not reveal specific details of a solution, nor should you show your written solution to anyone else. The only exception to this rule is work you’ve done with a lab partner while in lab. In this case, note who you’ve worked with and what parts were solved during lab.

0. Before final submission, make sure to fill out the README file.

1. Rank the following functions in ascending order of growth. You do not need complete proofs for this problem, but show your work if you want partial credit.

   - $f_1(n) = 2^n$
   - $f_2(n) = 3n(\log n)^3$
   - $f_3(n) = 100n^{5/4}$
   - $f_4(n) = 2^{\log n}$
   - $f_5(n) = 2^{2^n}$

   **Hint:** Remember the facts we saw in class. Ordering these functions is easier by combining known results/facts than deriving each comparison from scratch.

2. (Kleinberg and Tardos, 3.2) Give an algorithm to detect whether a given undirected graph contains a cycle. If the graph contains a cycle, then your algorithm should output one. Your algorithm should run in $O(m + n)$ time for a graph with $n$ vertices and $m$ edges.

3. (Kleinberg and Tardos, 3.9) There’s a natural intuition that two nodes that are far apart in a communication network—separated by many hops—have a more tenuous connection than two nodes that are close together. There are a number of algorithmic results that are based to some extent on different ways of making this notion precise. One way involves the subsceptibility of paths to the deletion of nodes.

   Suppose that an $n$-node graph $G = (V, E)$ contains two nodes $s$ and $t$ such that the distance\(^1\) between $s$ and $t$ is strictly greater than $n/2$. Show that there must exist some node $v$ not equal to either $s$ or $t$, such that deleting $v$ from $G$ destroys all $s \leadsto t$ paths. (In other words, the graph obtained from $G$ by deleting $v$ contains no $s \leadsto t$ paths).

   Give an algorithm with running time $O(n + m)$ to find such a node $v$.

---

\(^1\)Formally, define the distance between $s$ and $t$ as the length of the shortest path between $s$ and $t$. 
4. (Extra Credit.) This week, we saw an algorithm for testing bipartiteness which used BFS to color the vertices one of two colors. For a positive integer $k$, call a graph $k$-colorable if the vertices can be properly colored using $k$ colors. In other words, a bipartite graph is two-colorable. In this problem, you will investigate algorithms dealing with three-colorable graphs.

- Design and analyze an algorithm which takes as input a graph $G = (V, E)$ and returns \textsc{yes} if $G$ is three-colorable, and \textsc{no} otherwise.

- Design and analyze an efficient algorithm which takes as input a \textit{three-colorable} graph $G = (V, E)$ and colors the vertices of the graph using $O(\sqrt{n})$ colors. (Note: while the input graph \textit{is} three-colorable, it does not mean that we know what that coloring is!)