1. Consider a stable matching instance with $n = 4$ and the following preferences:

$m_1 : w_1 > w_2 > w_3 > w_4$

$m_2 : w_2 > w_3 > w_1 > w_4$

$m_3 : w_3 > w_1 > w_2 > w_4$

$m_4 : w_1 > w_2 > w_3 > w_4$

$w_1 : m_2 > m_3 > m_4 > m_1$

$w_2 : m_3 > m_4 > m_1 > m_2$

$w_3 : m_4 > m_1 > m_2 > m_3$

$w_4 : m_1 > m_2 > m_3 > m_4$

What is the output of the Gale-Shapley algorithm (G-S algorithm) on this instance?

(You can solve this question by hand or by code. However, you should know how to simulate the G-S algorithm by hand on small inputs.)

2. (2 points) In this exercise, we will prove that all executions of the G-S algorithm yield the same matching.

We say a woman $w$ is a valid partner of a man $m$ if there is a stable matching that contains the pair $(m, w)$. We say $w$ is the best valid partner of $m$ if $w$ ranks the highest (in $m$’s preference) among all valid partners of $m$. We write $\text{best}(m)$ for the best valid partner of $m$.

(a) Consider the following stable matching instance with $n = 2$:

$m_1 : w_1 > w_2$

$m_2 : w_2 > w_1$

$w_1 : m_2 > m_1$

$w_2 : m_1 > m_2$

Follow the above definitions and compute $\text{best}(m_1)$ and $\text{best}(m_2)$.

We will show that in any execution of the G-S algorithm, a man is never rejected by one of his valid partners. Because men propose in decreasing order of preference, the G-S algorithm must output the matching $S^* = \{(m, \text{best}(m))\}$. 
Suppose by contradiction that this may happen. Consider the first moment when some man \( m \) is rejected by one of his valid partners \( w \).

The rejection happened either because \( m \) proposed to \( w \) and was turned down, or because \( w \) broke her engagement to \( m \) in favor of a better proposal. Either way, at this moment \( w \) forms or continues an engagement with a man \( m' \) whom she likes more than \( m \).

Because \( w \) is a valid partner of \( m \), there exists a stable matching \( S \) containing the pair \((m, w)\). Suppose in this matching \( S \), \( m' \) is paired with \( w' \neq w \).

(b) Prove that the above assumptions would lead to the contradiction that \( S \) is not stable.

3. Order the following functions in ascending order of asymptotic growth rate. That is, if \( g(n) \) immediately follows \( f(n) \) in your list, then it should be the case that \( f(n) = O(g(n)) \).

\[
\begin{align*}
    f_1(n) &= n^2 \ln^3 n , \\
    f_2(n) &= 2^n , \\
    f_3(n) &= \ln n , \\
    f_4(n) &= n^{\ln n} , \\
    f_5(n) &= \ln \ln n , \\
    f_6(n) &= n^{4/3} , \\
    f_7(n) &= 2^{\sqrt{\log_2 n}} , \\
    f_8(n) &= n! .
\end{align*}
\]

4. Suppose we have two positive functions \( f \) and \( g \) such that \( f(n) = O(g(n)) \). Prove or disprove each of the following statement.

(a) \( \ln f(n) = O(\log g(n)) \).
(b) \( (f(n))^3 = O((g(n))^3) \).
(c) \( 2f(n) = O\left(2^{g(n)}\right) \).

5. The Floyd-Warshall algorithm is an algorithm for finding all-pair shortest paths in a weighted directed graph (with possibly negative edge weights but no negative cycles).

Give an asymptotically tight bound on the running time of the Floyd-Warshall algorithm on an input graph with \( n \) nodes.

(Assume that accessing an entry \( w[u][v] \) takes \( O(1) \) time. The function \( \min(x, y) \) returns the minimum of \( x \) and \( y \) which runs in \( O(1) \) time given \( x \) and \( y \).)
**Algorithm 1:** The Floyd-Warshall algorithm.

**Input:** An $n$-node graph with edge weights $w$.

**Output:** An $n \times n$ array $d$ where $d[u][v]$ is the minimum distance between $u$ and $v$.

for $u = 1$ to $n$
  for $v = 1$ to $n$
    if the edge $(u,v)$ exists then
      $d[u][v] \leftarrow w[u][v]$;
    else
      $d[u][v] \leftarrow +\infty$;
  
for $u = 1$ to $n$
  $d[u][u] = 0$;

for $k = 1$ to $n$
  for $i = 1$ to $n$
    for $j = 1$ to $n$
      $d[i][j] \leftarrow \min(d[i][j], d[i][k] + d[k][j])$;