Solutions of Assignment #1 part 1, Total points: 50
(Course: CS 401)

These are the first two problems for Assignment 1. **The remaining problems of Assignment 1 will be given out later.**

Problem 1 (20 points): Consider the stable matching problem as taught in class. Suppose that we have only three men, say m_1, m_2 and m_3, and only three women, say w_1, w_2 and w_3, with their corresponding preference lists. Suppose also that the matching m_1–w_1, m_2–w_2, m_3–w_3 is a stable matching. We make the following claim:

> in this case the matching m_1–w_1, m_2–w_3, m_3–w_2 can **never** be a stable matching.

Your task is to decide if our claim is true or false. For this purpose, do the following.

- Either prove the claim is indeed correct. Such a proof should work **no matter** what the preferences of the men and women are, as long as m_1–w_1, m_2–w_2, m_3–w_3 is a stable matching.
- Or, prove the claim made is wrong by giving a counter-example. The counter-example should provide the preferences lists of every man and woman, and show that for these preference lists **both** m_1–w_1, m_2–w_2, m_3–w_3 and m_1–w_1, m_2–w_3, m_3–w_2 are indeed stable matchings.

Solution: The following preference lists constitute a counter-example since both both m_1–w_1, m_2–w_2, m_3–w_3 and m_1–w_1, m_2–w_3, m_3–w_2 are stable matchings.

	first	second	third
m_1	w_1	w_2	w_3
m_2	w_2	w_3	w_1
m_3	w_3	w_2	w_1

	first	second	third
w_1	m_1	m_2	m_3
w_2	m_3	m_2	m_1
w_3	m_2	m_3	m_1

Problem 2 (30 points): Assume that you have two functions $f(n)$ and $g(n)$ such that $f(n) = O(g(n))$. Also, assume that $f(n) \geq 1$ and $\log_2 g(n) \geq 1$ for all n. For each of the following statements, decide whether you think it is true or false and accordingly give a proof (if true) or a counter-example (if false).

***(i)* (10 points)** $\log_2 f(n)$ is $O(\log_2 g(n))$

***(ii)* (10 points)** $2^{f(n)}$ is $O\left(2^{g(n)}\right)$.

***(iii)* (10 points)** $f(n)^3$ is $O\left(g(n)^3\right)$.

Solution:

(*i*) **True**.

$$f(n) = O(g(n)) \Rightarrow f(n) \leq c_1 g(n) \Rightarrow \log_2 f(n) \leq \log_2 c_1 g(n) = \log_2 c_1 + \log_2 g(n)$$

$$= \left(\frac{\log_2 c_1}{\log_2 g(n)} + 1 \right) \log_2 g(n) \leq (\log_2 c_1 + 1) \log_2 g(n) \leq c_2 \log_2 g(n)$$

Here we can choose any constant c_2 as long as $c_2 \geq \log_2 c_1 + 1$.

(*ii*) **False**. e.g. $f(n) = 2n$ and $g(n) = n$.
$2n = O(n)$, but $2^{2n} = 2^n \times 2^n$. We cannot find a constant c, such that $2^n \times 2^n \leq c2^n$ holds for any $n \geq n_0$

(*iii*) **True**.

$$f(n) = O(g(n)) \Rightarrow f(n) \leq cg(n) \Rightarrow f(n)^3 \leq (cg(n))^3 = c^3 g(n)^3$$