Question 1

For this problem, we will explore the issue of *truthfulness* in the Stable Matching Problem and specifically in the Gale-Shapley algorithm. The basic question is: Can a man or a woman end up better off by lying about his or her preferences? More concretely, we suppose each participant has a true preference order. Now consider a woman *w*. Suppose *w* prefers man *m* to *m'*, but both *m* and *m'* are low on her list of preferences. Can it be the case that by switching the order of *m* and *m'* on her list of preferences(i.e., by falsely claiming that she prefers *m'* to m) and running the algorithm with this false preference list, *w* will end up with a man *m"* that she truly prefers to both *m* and *m'*?(We can ask the same question for men, but will focus on the case of women for purposes of this question.)

Resovle this question by doing one of the following two things: (a) Give a proof that, for any set of preference lists, switching the order of a pair on the list cannot improve a woman's partner in the Gale-Shapley algorithm; or

(b) Give an example of a set of preference lists for which there is a switch that would improve the partner of a woman who switched preferences.

There maybe a switch that would improve the partner of a woman who switched preferences.

Consider the following instance. As show in Table 1.1 and Table 1.2, we can easy get the stable match which is 'A-Y, B-Z, C-X'.

Table 1.1 Men's preference table

	1st	2st	3st
Xavier	A	C	B
Yancey	A	B	C
Zeus	B	A	C

Table 1.2 Women's preference table

	1st	2st	3st
Amy	Z	Y	X
Bertha	Y	X	Z
Clare	Z	X	Y

Next, we change the preference of Amy. We exchange the indexes of 'Y' and 'X' in Amy's preference and get Table 1.3.

Table 1.3 Women’s preference table(changed)

	1st	2st	3st
Amy	Z	X	Y
Bertha	Y	X	Z
Clare	Z	X	Y

We can find that after the change, the final result is ’A-Z, B-Y, C-X’ which indicates that Amy matches a better mate Zeus.

Question 2

Suppose you have algorithms with the six running times listed below.(Assume these are the exact number of operations performed as a function of the input size *n*.) Suppose you have a computer that can perform 10^{10} operations per second, and you need to compute a result in at most an hour of computation. For each of the algorithms, what is the largest input size *n* for which you would be able to get the result within an hour?

(a) n^2

(b) n^3

(c) $100n^2$

(d) $n\log n$

(e) 2^n

(f) 2^{2^n}

Let $X \in \{(a),(b),(c),(d),(e),(f)\}$ be one of the six running times. We have $X \leq 3600 \times 10^{10}$. We calculate the result in Table 2.1.

Table 2.1 Limited value of n

	Running times	Compute performance (s^{-1})	Limited time(s)	n
(a)	n^2	10^{10}	3,600	6×10^6
(b)	n^3	10^{10}	3,600	33,019
(c)	$100n^2$	10^{10}	3,600	6×10^5
(d)	$n\log n$	10^{10}	3,600	906,316,482,853
(e)	2^n	10^{10}	3,600	45
(f)	2^{2^n}	10^{10}	3,600	5

Question 3

Take the following list of functions and arrange them in ascending order of growth rate. That is, if function $g(n)$ immediately follows function $f(n)$ in your list, then it should be the case that $f(n)$ is $O(g(n))$.

1. $f_1(n) = n^{2.5}$
2. $f_2(n) = \sqrt{2n}$
3. $f_3(n) = n + 10$
4. $f_4(n) = 10^n$
5. $f_5(n) = 100^n$
6. $f_6(n) = n^2 \log n$
7. $g_1(n) = 2^{\sqrt{\log n}}$
8. $g_2(n) = 2^n$
9. $g_4(n) = n^{4/3}$
10. $g_3(n) = n(\log n)^3$
11. $g_5(n) = n^{\log n}$
12. $g_6(n) = 2^{2^n}$
13. $g_7(n) = 2^{n^2}$

It is easy to get the following basic sequence showed in Table 3.1.

Table 3.1 Part of the ranking

2	$f_2(n) = \sqrt{2n}$
3	$f_3(n) = n + 10$
9	$g_4(n) = n^{4/3}$
1	$f_1(n) = n^{2.5}$
8	$g_2(n) = 2^n$
4	$f_4(n) = 10^n$
5	$f_5(n) = 100^n$
13	$g_7(n) = 2^{n^2}$
12	$g_6(n) = 2^{2^n}$

Then we compare the others with some pivotal functions in Table 3.1.

Consider $f_6(n) = n^2 \log n$, we can easily know that $f_6(n) = O(f_1(n))$ and $g_4(n) = O(f_6(n))$.

Maybe, it's difficult to find the position of $g_1(n)$. We compare $g_1(n)$ with $f_2(n)$. Let $m = \sqrt{\log n}$, then we have $n = 2^{m^2}$. As show in Equation 3.1. So we have $g_1(n) = O(f_2(n))$.

$$\lim_{n\to\infty} \frac{g_1(n)}{f_2(n)} = \lim_{m\to\infty} \frac{2^m}{\sqrt{2} \cdot 2^{\frac{1}{2}m^2}} = 0 \tag{3.1}$$

It is obviously that $f_3(n) = O(g_3(n))$. Now we compare $g_3(n) = n(\log n)^3$ with $g_4(n) = n^{4/3}$. Let $m = \log n$, then $n = 2^m$. And we can get Equation 3.2.

$$\lim_{n\to\infty} \frac{g_3(n)}{g_4(n)} = \lim_{m\to\infty} \frac{m^3}{2^{\frac{1}{3}m}} = 0 \tag{3.2}$$

Therefore, $g_3(n) = O(g_4(n))$.

Finally, we assure where $g_5(n) = n^{\log n}$ should be placed. It is easy to know that $f_1(n) = O(g_5(n))$. Owing to the fact that $g_5(n) = n^{\log n} = 2^{\log(n^{\log n})} = 2^{(\log n)^2}$, we guess $g_5(n)$ is between $f_1(n)$ and $g_2(n)$. Let $m = \log n$, then $n = 2^m$. And we can get Equation 3.3.

$$\lim_{n\to\infty} \frac{g_5(n)}{g_2(n)} = \lim_{m\to\infty} \frac{2^{m^2}}{2^{2^m}} = 0 \tag{3.3}$$

So we have $g_5(n) = O(g_2(n))$.

We summarize the final result in Table 3.2.

Table 3.2 Rank of functions

7	$g_1(n) = 2^{\sqrt{\log n}}$
2	$f_2(n) = \sqrt{2n}$
3	$f_3(n) = n + 10$
10	$g_3(n) = n(\log n)^3$
9	$g_4(n) = n^{4/3}$
6	$f_6(n) = n^2 \log n$
1	$f_1(n) = n^{2.5}$
11	$g_5(n) = n^{\log n}$
8	$g_2(n) = 2^n$
4	$f_4(n) = 10^n$
5	$f_5(n) = 100^n$
13	$g_7(n) = 2^{n^2}$
12	$g_6(n) = 2^{2^n}$