Instructions COMP 3804 Assignment 1 Due on Wednesday, October 7th (In the class just before the start of the class - Any late submissions will be awarded a Mark of 0, although they will be corrected, as we will most likely discuss the solution of the problems in that class - no solutions will be posted!!).

Please write clearly and answer questions precisely. Please cite all the references (including web-sites, names of friends, etc.) which you used/consulted as the source of information for each of the questions.

Please put your answers in order; Answer for Problem 1 followed by Problem 2 etc. Moreover please do provide your name and student number. It will be best that you keep all your marked Assignments till the end of the course, since sometimes the marks get failed to be recorded.

All questions carry equal marks.

Consider the Proposal Algorithm that was discussed in the class. Its outlined below:
Input: A list of n men and n women, where each man has an ordered list of n women (a permutation) whom he will like to marry. Similarly, each woman has an ordered list (a permutation) of n men whom she will like to marry.
Output: A set of n pairs forming a stable perfect matching. Each pair consists of a man and a woman.

```
Proposal-Algorithm (M,W)
1. While there exists an unmarried man m and
                       has not proposed to all the women do
2.      m proposes to the most preferred woman w on his list
          whom he has not proposed so far.
3.      if w is not married then w marries m (end if)
4.      if w is married to m' but prefers m over m' then
5.         w divorces m' and marries m
        (end if)
   (endwhile)
```

1. Prove or Disprove: The output of this algorithm is always the same and is independent of in which order the men are picked up in Step 1. In other words the above algorithm produces the same set of stable matchings and is independent of the choice of men m in Line 1. (Recall $(m, best(m))$).

2. List all the data structures (including the operations) that you will use to implement the Proposal Algorithm. (e.g. Stacks/Arrays/...). What is the complexity of your algorithm? Analyze in detail.

3. Consider the stable matching problem, where certain man-woman pairs are forbidden. Each man m ranks all the women, except the ones which are forbidden, and same with each woman. We can assume that the set F consists of the forbidden pairs, $F \subseteq M \times W$, where M is the set of n-men, and W is the set of n-women. Modify the definition of stable matchings to take into account the forbidden set F. Modify the above proposal algorithm to compute a set of stable matchings. Prove that your algorithm terminates and produces stable matchings.

4. Let $p(n) = a_d n^d + a_{d-1} n^{d-1} + \cdots a_1 n + a_0$, where $a_d > 0$, be a d-degree polynomial in n. Also $a_0, \cdots, a_d$ are constants. Let k be a positive integer. Prove the following:

   (a) If $k \geq d$, then $p(n) = O(n^k)$.
   (b) If $k \leq d$, then $p(n) = \Omega(n^k)$.
   (c) If $k = d$, then $p(n) = \Theta(n^k)$.
   (d) If $k > d$, then $p(n) = o(n^k)$.

   (For this question please review the definitions of O, Θ, Ω, o, and prove each of the above statements by showing the appropriate constants c, c_1, c_2, n_0; See Chapter 3 in the book.)

5. Solve the recurrence relation

$$T(n) = T(xn) + T((1-x)n) + cn$$

   in terms of x and n where x is a constant in the range $0 < x < 1$. Is the asymptotic complexity the same when $x = 0.5, 0.1$ and 0.001? What happens to the constant hidden in the $O()$ notation. (Hint: Please review the last part of Section 4.4 in the 3rd edition of the text-book or Section 4.2 from the 2nd edition.)

6. Show that the following recurrence evaluates to $O(n)$.

$$T(n) \leq \begin{cases} T(\lceil n/5 \rceil) + T(\lceil 7n/10 \rceil) + O(n) & \text{if } n > 140, \\ O(1) & \text{if } n \leq 140. \end{cases}$$

   (This one is in the book!)

7. Analyze the recurrence relation

$$T(n) = \sqrt{n}T(\sqrt{n}) + n.$$

(This is called as the rootish-divide-and-conquer and for example plays an important role in parallel computing.)

8. Assume that you have a sorted array of size n containing real numbers. State the recurrence relation for time for performing binary search on this array. Analyze it and show that it evaluates to $O(\log n)$; Is it $\Theta(\log n)$ as well?

9. Suppose you need to choose between the following algorithms which solves the same problem:

   (a) Algorithm A solves the problem by dividing it into 5 subproblems of half of the size, recursively solves each of them, and combines the solution in linear time.

   (b) Algorithm B solves the problem of size n by recursively solving two subproblems of size $n - 1$ and then combining the solutions in constant time.

   (c) Algorithm C solves the problem of size n by dividing it into 9 subproblems of size $n/3$ each, recursively solving each of them, and then combining the solution in $O(n^2)$ time.

   What are the running times of each of these algorithms and which one will you choose?

10. Solve the following recurrence $T(n) = 2T(n/2) + O(n \log n)$, where $T(n)$ is a constant for all values of $n \leq 4$.

11. (Bonus Problem) Can you devise a proposal algorithm that is unbiased? (Look up on Google and find an appropriate reference!)

12. (Bonus Problem) V. Pan has discovered a way to multiply two $70 * 70$ matrices using only 143640 multiplications. Ignoring the additions, what will the asymptotic complexity of Pan's algorithm for multiplying two $n * n$ matrices? Is it better than Strassen's? Justify your answer.