UBC, CPSC 320: Intermediate Algorithm Design and Analysis (Winter 1 2015/2016)

Assignment 1

Due: September 25, 2015 at 3.30p

1. **(Alternative algorithm for the stable matching problem)** In class, one algorithm suggested for the stable matching problem was to first pair men and women in arbitrary way, and then to find unstable pairs, and fix them by swapping partners. Suppose that we are given three women and three men with the following preference lists:

   $$\begin{array}{llcll} m_1: & w_1, w_3, w_2 & \quad & w1: & m_2, m_1, m_3 \\ m_2: & w_3, w_1, w_2 & & w2: & m_1, m_2, m_3 \\ m_3: & w_1, w_2, w_3 & & w3: & m_1, m_2, m_3 \end{array}$$

   Give an example of an initial pairing, and a sequence of fixes of unstable pairs that will never reach a stable matching by ending in an infinite loop.

2. **(Property of Gale-Shapley algorithm)** Let S^* be the solution return by GS algorithm from the lecture/textbook. Let $S^\triangle$ be the solution returned by the modified GS algorithm in which the women do the proposing.

   Prove that if $S^* = S^\triangle$ for an instance of the Stable Matching Problem then there is only one solution for this instance.

   (You can use any theorems mentioned at the lecture/textbook.)

3. **(Using GS algorithm to solve another problem)** Solve Exercise 6 from Chapter 1 in the textbook.

   It's recommended that instead of trying to modify the GS algorithm, you try to use it as a black box in your algorithm. Remember to prove the correctness of your algorithm.

4. **(Algorithm Analysis)**

   (a) Determine the asymptotic tight bound (Θ) on the worst-case running time of the algorithm below that determines whether an array of size n contains duplicates.

   ```
   function HasDuplicates($A$)
       for $i \leftarrow 1$ to $length(A) - 1$ do
           for $j \leftarrow i + 1$ to $length(A)$ do
               if $A[i] = A[j]$ then
                   return $True$
               end if
           end for
       end for
       return $False$
   end function
   ```

   Remember to prove the asymptotic tight bound, you need first show the asymptotic upper bound and then find an instance for which the running time of this algorithm will be lower bounded by the same function.

   (b) Is this algorithm optimal? Either show that any algorithm solving this problem will have the worst-case running time lower bounded by the same function, or propose a more efficient algorithm.

5. **(Bonus)** How long did it take you to complete this assignment (not including any time you spent revising your notes before starting)?