Problem Set 2 **Due: Feb 27** **cs2me3 Winter 2006**

Instructions

1. You are encouraged to work in groups of two. If you cannot find a partner, you can work alone.
2. Please submit one copy of the assignment; if you are working with a partner, both names should appear on the assignment.
3. For **Part A** of the assignment, you must submit an electronic copy of your Java application via WebCT (by the time of the lecture on the due date of the assignment).

Part A

Write a Java application which solves the Stable Marriage Problem, by implementing the Gale-Shapley algorithm presented in class.

Your program should work as follows: given an ASCII text file as input, `preferences.txt` containing the preferences of the boys and girls, it outputs a stable matching.

The input file should be of the form

```
3
2<1<3 3<1<2
2<3<1 1<2<3
3<2<1 3<2<1
```

i.e., the first line should give the size of $|B| = |G|$, and then two columns should follow, giving (in order) the preferences of the boys and girls.

The output (to the standard output) should be:

```
1-2
2-1
3-3
```

(Here is the justification: in stage 1, we obtain $\{(b_1, g_2)\}$. In stage 2, b_2 wants g_2, but g_2 prefers b_1, so b_2 goes for his second choice, which is g_3, so we have $\{(b_1, g_2), (b_2, g_3)\}$. In stage 3, b_3 wants g_3, who is engaged, but g_3 prefers b_3 to current b_2, so b_3 gets g_3, and now b_2 chooses g_1, so we have $\{(b_1, g_2), (b_2, g_1), (b_3, g_3)\}$.)

Part B

1. Exercises 2.1, 2,2, 2.3, and 2.4 in the notes.

   Solutions: 2.1 b_{s+1} proposes to the g's according to his list of preference; a g ends up accepting, and if the g who accepted b_{s+1} was free, she is the new one with a partner. Otherwise, some $b^* \in \{b_1, \ldots, b_s\}$ became disengaged, and we repeat the same argument. The g's disengage only if a better b proposes, so it is true that $p_{M_{s+1}}(g_j) <^j p_{M_s}(g_j)$.

   2.2 If some b wants to propose to the same g for a second time, it means that they already once disengaged, and since the g's disengage only when they get a better offer, it means that now g still has a better partner than b, and so even if b proposed, he would be rejected. Therefore, it is not necessary for the b's to propose twice to the same g.

   2.3 Suppose that we have a blocking pair $\{b, g\}$ (meaning that $\{(b, g'), (b', g)\} \subseteq M_n$, but b prefers g to g', and g prefers b to b'). Either b came after b' or before. If b came before b', then g would have been with b or someone better when b' came around, so g would not have become engaged to b'. On the other hand, since (b', g) is a pair, no better offer has been made to g after the offer of b', so b could not have come after b'. In either case we get an impossibility, and so there is no blocking pair $\{b, g\}$.

   2.4 To show that the matching is boy-optimal, we argue by contradiction. (Let g *is an optimal partner for* b mean that among all the stable matchings g is the best partner that b can get.)

   We run the Gale-Shapley algorithm, and let b be the first boy who is rejected by his optimal partner g. This means that g has already been paired with some b', and g prefers b' to b.

   Furthermore, g is at least as desirable to b' as his own optimal partner (since the proposal of b is the first time during the run of the algorithm that a boy is rejected by his optimal partner).

   Since g is optimal for b, we know (by definition) that there exists some stable matching S where (b, g) is a pair. On the other hand, the optimal partner of b' is ranked (by b' of course) at most as high as g, and since g is taken by b, whoever b' is paired with in S, say g', b' prefers g to g'.

   This gives us an unstable pairing, because $\{b', g\}$ prefer each other to the partners they have in S.

   To show that the Gale-Shapley algorithm is girl-pessimal, we use the fact that it is boy-optimal (which we just showed). Again, we argue by contradiction. Suppose there is a stable matching S where g is paired with b, and g prefers b' to b, where (b', g) is the result of the Gale-Shapley algorithm.

   By boy-optimality, we know that in S we have (b', g'), where g' is not higher on the preference list of b' than g, and since g is already paired with b, we know that g' is actually lower. This says that S is unstable since $\{b', g\}$ would rather be together than with their partners.