Prob. 1	Prob. 2	Prob. 3	Prob. 4

Problem 1.

Problem 2.

Problem 3.

1. Strong instability
   Yes, there always exists a perfect matching without any strong instability.
   An algorithm in polynomial-time that can guarantee such a matching is actually the ”man proposes, woman disposes” we've seen in class.
   This algorithm has a wort-case number of rounds of $n^2 - 2n + 2$ so it's running in polynomial-time.
   In the case of ties, we may end up with m prefering w than its final matching and w having m at the top of her ranking. But it means that the final matching of w is a man she likes as much as m not less otherwise it means than m has never proposed to w which contradicts the way the algorithm works.
   Thus with the ”man proposes, woman disposes” algorithm we can never have a strong instability.

2. Weak instability
   First we can see that the algorithm used for avoiding strong instabilities doesn't work for weak instability. The explanation of the correctness used before actually shows a weak instability.
   But anyway there isn't any algorithm that can avoid weak instabilities for sure. For instance if you have m_1 and m_2 both prefering w_1 than w_2 and w_1 liking m_1 and m_2 equally (we don't even need to know the preferences of w_2). The two only possible matching without even taking rankings into account would be:

   (a) $m_1 \longleftrightarrow w_1$ and $m_2 \longleftrightarrow w_2$
   (b) $m_1 \longleftrightarrow w_2$ and $m_2 \longleftrightarrow w_1$

   In (a) we can see that there's a weak instability because m_2 prefers w_1 to his matching and w_1 likes m_1 and m_2 equally.
   In (b) there is also a weak instability because m_1 this time prefers w_1 to his matching and w_1 likes m_1 and m_2 equally.
   Thus there is no solution without a weak instability.

Problem 4.