Assignment 3
CSCI 3110: Design and Analysis of Algorithms
Due June 5, 2015

Banner ID: ________________________ Name: ________________________________

Banner ID: ________________________ Name: ________________________________

Banner ID: ________________________ Name: ________________________________
**Question 1 (15 marks)** You are organizing a soccer match with your friends. Since you have played together many times before, each of you knows how well they play on the same team with everybody else. As a result, each of you may express one of the following two preferences for any other member \( y \) of your group:

- I would rather not play on the same team with \( y \).
- I really want to play on the same team with \( y \).

Note that you may not care whether you play on the same team with \( y \); in that case, you don't express any preference concerning \( y \). Your goal is to figure out whether there is a way to form two teams so that

1. all preferences of every member of your group are satisfied and
2. there are at least 11 players on each team.

Develop an algorithm that answers this question. Your algorithm should run in \( O(n + m) \) time, where \( n \) is the number of people in your group and \( m \) is the number of preferences that were expressed. Argue that the algorithm is correct and that its running time is indeed \( O(n + m) \). You may assume that \( n \geq 22 \) because otherwise it is obviously impossible to have at least 11 players on each team.

**Hints:**

- First worry about testing whether it is possible at all to form two teams that satisfy all the preferences, regardless of the sizes of the two teams. Then see whether you can swap groups of players between the teams so that the preferences remain satisfied and you end up moving players from the bigger team to the smaller one until each team has at least 11 players. This latter step should be easy if there are lots of “swappable” groups of players. If there are only few such groups, you will need to use an exponential-time algorithm. However, the input size to this algorithm will be bounded by a constant in this case, so an exponential running time in the input size is still constant and thus in \( O(n + m) \).

- For testing whether it is at all possible to satisfy all the preferences, the algorithm for testing bipartiteness of a graph is a very good starting point. You should have no trouble to convince yourself that the problem of satisfying the constraints that state that a player does not want to play on the same team with another player can easily be reduced to testing whether an appropriately defined graph is bipartite. The challenge thus is to figure out whether you can adapt the bipartiteness testing algorithm so it also takes preferences to play on the same team with another player into account.