CSE 417: Homework 2

Name: Qingchuan Hou

Student ID: 2127437

UWNetID: qhou

Problem 6:
Mikileaks has posted n (thousands of) blurry images purportedly stolen from a secrect government database of UFO sitings. You believe each potentially belongs to one of two different groups, perhaps Romulans vs Dragons (R and D for simplicity), but it’s hard to directly label any one image in isolation. Instead, you study each pair i and j side-by-side; label the pair (i, j) either “same” (meaning you believe them both to come from the same group) or “different” (they represent different groups). As a third alternative, you may offer no opinion on a given pair; call the pair ambiguous.
Given the collection of n images, as well as a collection of m judgments (either “same” or “different” for the non-ambiguous pairs), you’d like to know if this data is consistent with the idea that each image is from one of only two groups. More concretely, we’ll declare the m judgments to be consistent if it is possible to label each image either R or D in such a way that for each pair (i, j) labeled “same,” it is the case that i and j have the same label; and for each pair (i, j) labeled “different,” it is the case that i and j have opposite labels. For example, if 1 and 2 are “same”, and 1 and 3 are “same”, but 2 and 3 are “different”, then there is no consistent labeling, whereas if 2 and 3 are “same”, then there is a consistent labeling.
Give an algorithm with running time O(m + n) that determines whether the m judgments are consistent.

```
initial state: all "image" undiscovered
for "image" = 1 to n
if undiscovered and have judgments then
mark state('image) == discovered
if have judgments then
for other have pair(judgements) with image
if undiscovered:
if same: label same label(P or R) and mark discovered
if different: label same label(P or R) and mark discovered
if discovered:
if same label and same judgement or different judgement with different lable: label consistent
else : label inconsistent
```

This algorithm will search every edge for each node one time. Each edge is touched twice. So the runtime is O(m+n)