HW12 - 5.26 + 5.28

5.26) For set of variables $x_1 \dots x_n$ & input m of constraints of for "$x_i = x_j$" or "$x_i \neq x_j$", give efficient algorithm to decide if all constraints can be satisfied.

```
checkconstraints(m, n)   m = variables, n = constraints
makeset for all variables to x_n in n → {x1}, {x2}, ... {xn} disjoint set
for all constraints, c in m  * Iterate through each constraint, assuming prework
    left = left side variable of c to make sure all equalities core first has been done *
    right = right side variable of
    if c is equality
        Union(left, right) * Add two variables into same disjoint set to create equivalence classes
    if c is disequality
        if find(left, right)==True * If two variables that are supposed to not be equal
            return False -  are in the same disjoint set, the constraint is violated: return false
return True
```

S.28) Alice's party - n people to choose from, list of pairs who know each other. Invitee must 1) have at least 5 people they know & 2) 5 people they don't know

```
choose invitees(n, pairs)
inviteeset = make set of dupl objects of {person, count=0} for all people in n
while person removed is True:  ← Person removed = False
    for all pairs in pairs
        increment count for person1 & person2 of pair
    for all invitees in inviteeset:
        if count < 5  * does not know
            remove invitee from inviteeset
            person removed = True  ← remove pair containing person from pairs
        else if count > n-5  * knows too many people
            remove invitee from inviteeset
            person removed = True
            remove pairs containing person from pairs
return inviteeset
```