Matt Forbes
October 7
Homework 1

1 Problem One

1.1

The minimum number of socks that need to be drawn to guarantee a pair of any color is 4. You could possibly have 3 socks that do not make a pair, but on the next draw, one of the socks will become a match.

1.2

Likewise, the minimum number of socks from a drawer of n different color socks (assuming there is at least 2 of each color) is n+1. For the same reason that you can have no match on the nth draw, but on the n+1th draw, one sock will match.

2 Problem Two

Prove by induction that $F_{n+k} = F_k F_{n+1} + F_{k-1} F_n$

2.1 Basis

$$\begin{aligned} \text{let n=0, k >= 1} \\ F_{0+k} = F_k F_1 + F_{k-1} F_0 \\ F_k = F_k(1) + F_{k-1}(0) \\ F_k = F_k \end{aligned}$$

2.2 Inductive Hypothesis

For all $n >= 1, c >= 0$, assuming the equation is true for $n - 1$ then it is also true for n.

2.3 Proof

$$\begin{aligned}
F_{n+c} &= F_c F_{n+1} + F_{c-1} F_n \\
F_{(n-1)+(c+1)} &= F_c F_{n+1} + F_{c-1} F_n \\
F_{c+1} F_n + F_c F_{n-1} &= F_c (F_{n-1} + F_n) + F_{c-1} F_n \\
(F_{c-1} + F_c) F_n + F_c F_{n-1} &= F_c F_{n-1} + F_c F_n + F_{c-1} F_n \\
F_{c-1} F_n + F_c F_n + F_c F_{n-1} &= F_{c-1} F_n + F_c F_n + F_c F_{n-1}
\end{aligned}$$

3 Problem Three

Prove $\sum_{i=0}^{n} \binom{n}{i} = 2^n$ with induction and given identity.

3.1 Basis

$$\begin{aligned}
\sum_{i=0}^{1} \binom{1}{i} &= \binom{1}{0} + \binom{1}{1} \\
&= 1 + 1 \\
&= 2^1
\end{aligned}$$

3.2 Inductive Hypothesis

For all $n > 1$, assuming the sum is true for $n - 1$, it is also true for n.

3.3 Induction

$$\begin{aligned}
\sum_{i=0}^{n}\binom{n}{i} &= \binom{n}{0}+\sum_{i=1}^{n}\binom{n}{i}\text{[pull first value from sum]}\\
&= 1+\sum_{i=1}^{n}\binom{n-1}{i}+\binom{n-1}{i-1}\text{[given identity]}\\
&= 1+\sum_{i=1}^{n}\binom{n-1}{i}+\sum_{i=1}^{n}\binom{n-1}{i-1}\text{[break up sum]}\\
&= 1+[(\sum_{i=0}^{n}\binom{n-1}{i})-\binom{n}{0}]+\sum_{i=0}^{n}\binom{n-1}{i}\text{[change sum limits]}\\
&= 1+[2^{n-1}-1]+2^{n-1}\text{[inductive hypthosis]}\\
&= 2*2^{n-1}\\
&= 2^{n}
\end{aligned}$$

4 Problem Four

4.1 Measure 1: Simplicity of Plan

I had a hard time quantifying the efficiencies of this problem, but I think that since we are working with people, not computers, the ease of carrying out the plan can be a large factor. A complex plan might end up bringing everyone together faster, but if it is too complicated, some of the troops may screw it up. So this plan is optimized to provide easy instructions to the troops.

n = the number of troops in the team.
x_i = soldier i, where i = 1...n
d_i = distance assigned to x_i, where i = 1...n
let $d_i = (-1)^i i$, where i = 1...n

Walking Plan: (for x_i)

1. Walk d_i meters (where -d is to the left).
2. If you reach another soldier, then stay together as a group, choosing your new d to be the greatest d of the group.
3. Your new d is now $-2d$

4. If you do not have all members of the team in your group, do step 1 again.

Why this works
This will always work because the soldier with the largest initial d will end crossing the path of all the other soldiers at one point because he will be going the furthest in both directions. The other soldiers are also moving, and will hopefully group up faster rather than just waiting to be picked up by the furthest mover.

4.2 Measure 2: Fastest Rendezvous of Team

The obvious optimization is the time it takes to get the whole group together in the shortest amount of time. This plan is quite a bit more complex, but it should speed up the grouping by quite a bit. It uses a similar algorithm as the first plan, but handles the collision of two soldiers quite a bit differently.

n = the number of troops in the team.
x_i = soldier i, where i = 1...n
d_i = distance assigned to x_i, where i = 1...n
let $d_i = (-1)^i i$, where i = 1...n

Walking Plan: (for x_i)
Case 1 (Default):

1. Walk d_i meters (where -d is to the left).
2. If you meet another soldier, you are now a group. Both of your d values are now equal to the greatest of the two, or to your teammates value if he is in a group. You also keep your original direction. Your new goal is to find all the soldiers on your side of where you met. Go to case 2.
3. If you didn't meet another soldier, then your d is now $-2d$.
4. Go to case 1.

Case 2 (Grouped with one partner):

1. You and your "partner" are now responsible for reporting the number of teammates you each find on the side of the meeting point until you have found the entire team.
2. Walk d meters (where -d is to the left).

3. If you meet another soldier, you have completed a group. And your new job is simply to relay the number of teammates currently found on your side of the field. Go to case 3.
4. If you didn't meet another solder, then walk $-d$ meters back to where you first collided with a member, and meet them there.
5. Add up the total number of soldiers found on both sides, and if all members are found then go to case 4.
6. Otherwise, your d is now $2d$, go to case 2.2.

Case 3 (Grouped on both sides):

1. Walk $-d$ meters back where you had your first collision, and meet your "partner" there.
2. Add up the total number of soldiers on both sides (including soldiers your new teammate found), and if all members are accounted for go to case 4.
3. Otherwise, your d is now $2d$.
4. Walk d meters.
5. go to case 3.

Case 4 (Final):

1. If there are no teammates in the positive direction, then stop and wait for the group to come to you.
2. Otherwise, if there are no teammates in the negative direction, just walk in the positive direction until you meet up with the rest of the team.
3. Otherwise, walk d meters and tell them that everyone is accounted for, then walk in the positve direction until you meet up with the rest of the team.

Why this works
This is a similar routine as the first plan. The only thing is that rather than alternating directions after you find a teammate, you are smart about it and split the problem in half. You look on one side, and they look on the other. This will always work, because everyone starts by probing for nearby members, then scouts on one side of that collision. By moving further and further in that direction everytime, they are guaranteed to find everyone else on that side of them.