Lecture 13: 5/14/2009

Announcements: Ps7 out. Ps6, Midterm solutions out
Midterm back (median 75 about a low B)

```
1. Infinite Sets, Review/wrap-up
---------------------------
```

We showed that the Integers and Rationals (Z and Q) though infinite, are **countable,** in a sense the lowest level of infinity. The reals or P(N) are uncountable, but at the next level, while P(R) or P(P(N)) are at a higher level of infinity.

```
2. The Pigeonhole Principle (Counting, 5.4)
---------------------------

If N pigeons roost in n holes, N>n, then some two pigeons
share a hole.

Restated:
if f: A -> B where A and B are finite sets, |A|>|B|,
the f is NOT injective.


Ex 0.  Any room with 3 or more people has some two of the same
gender.

Ex 1.  20 people at a party, some two have the same number of
friends (we assume “friends” is symmetric”.
       number of friends
       proof: 0..18   or 1..19

Ex 2: Given five points inside the square whose side is of
length 2, prove that
      two are within \sqrt{2} of each other.

       Soln: divide square into four 1 x 1 cells.  Diameter of
each cell = \sqrt{2}

Ex 3: In any list of 10 integers, a_1, ..., a_10, there's a
string of
      consecutive numbers whose sum is divisible by 10.

      Consider a_1
```

$a_1 + a_2$
...
$a_1 + a_2 + ... + a_{10}$

If any of these divisible by 10: done.
Otherwise, each is congruent to 1,..., 9 mod 10, so two are congruent
to the same thing, eg,

$$a_1 + a_2 + a_3 = 6 \pmod 5$$
$$a_1 + a_2 + a_3 + a_4 + a_5 = 6 \pmod 5$$

But then
$$a+4 + a_5 = 0 \pmod{10}$$

Ex 4, In any room of 6 people, there are 3 mutual friends or 3 mutual strangers (Ramsey theorem)

Remove person 1   5 people left.
Put into two pots: friends with 1, non-friends with 1.
One has at least three people.
If three friends of 1: Case A: some two know each other: DONE
Case B: no two know each other: DONE

If three non-friends of 1: Case C: Some two of these 3 don't know each other -> those two and 1 are 3 mutual stangers.

Case D: Each pair know each other, so we have 3 mutual friends.

Difficult Puzzle (not discussed in class): What is the minimum number of people that must assemble in a room such that there will be at least n friends or n non-friends: R(n,n)

R(4,4) = 18 (1955)
R(5,5) = ?? open!!! known to be between 43 (1989) and 49 (1995)
R(10,10) =?? open and not tightly determined at all: range 798 (1986)- 23,556 (2002)

The above example uses the

"Strong form of Pigeonhole principle:

If f: A -> B, A and B finite sets,
then some point in B will have at least
\lceil |A|/|B| \rceil preimages under f

Book form: if kn+1 pigeons and in n holes, some hole has at least k+1 pigeons (for k a positive integer).

----

3. Induction and recursion and analysis recursive programs

EXAMPLE 1. Sam's Dept Store sells enveloped in packages of 5 and 12.

Prove that, for any n>= 44, the store can sell you exactly n envelopes.

```
Try it:  44 = 2(12) + 4(5)
         45 =         9(5)
         46 = 3(12) + 2(5)
         ?...?
```

SUPPOSE: it is possible to buy k envelopes for some k\ge44.
SHOW: it is possible to buy k+1 envelopes

If purchasing at least seven packets of 5, trade them for three packets of 12:

```
  7(5) -> 3(12)
  35       36
```

If <7 packets of 5, ie <=6 packets of 5, so at most 30 of the envelopes are in packets of 5; so there are >= 44-30 = 14 envelopes
being bought in packets of 12, so >= 2 two packets of twelve.
So take two of the packets of 12 and trade them for 5 packets of 5:

```
  2(12) -> 5(5)
   24       25
```

This is an example of an **exchange argument**, often used on proofs of solution methods.

Principle of mathematical induction (review)

```
-----------------------------------
To prove a proposition P(n) for all integers n>= n_0:
1) Prove P(n_0)    (Basis)
2) Prove that P(k) --> P(k+1) for all k > n_0    (inductive
step)
    (Inductive hypothesis)

(initially make n_0=1)
```

```
-------------------------------------------------
Example : Binary Search
-------------------------------------------------
```

Suppose we have a sorted list of n numbers in an array A[1..n] and we want to find if a number X is in this list. Last lecture we describe an O(n) linear search algorithm, here we give an improved method.

```
Binary-Search(X,A, low, high):Looks for X in A[low..high]
Start
(0<low ≤ high ≤ n)
If(i>j) Return("not found"); // (range of A is empty)
Middle ←⌊(i+j)/2⌋;
If(A[Middle] = X) Return (Middle)
If (A[Middle] < X) low ← Middle+1 // X not in A[1..Middle]
If (A[Middle] >X) high ← Middle-1 // X not in A[Middle..high]
Binary-Search(X,A,low,high);
End;
```

To start call Binary-Search(X,A,1,n).

Example: A: 3,5,10, 16, 30, 60, 90, X=10
Low:1, high: 7
Middle: 8/2=4, A[4]= 16 > 10, so high ←middle-1=3

Middle: 1+3/2= 2, A[2]=5 < 10, so low ← middle +1 =3

Middle: 3+3/2=3, A[3] = 10=X, return 3.

Suppose X = 12? Only differs in last step, 10 < X, so low ←4.
Next call is to BS(X,A, 4,3), so 4>3 and return "not found"

Analysis of Run time:

We can assume that each iteration of the program (not including the recursive call in the last line), takes constant time (that is, it does not depend on n). Thus to get the big Theta run time, we only need to count the number of calls.

Let $T(n)$ be the maximum number of calls for an array of size n.

Finish next time (and proof of correctness).