Midterm 2, CS161, Spring '12, Section 1

Consider the following declaration:

```
class Node
{
   private Object item;
   private Node next;

   public Object getItem()
   {
      return item;
   }

   public void setItem(Object ob)
   {
      item = ob;
   }

   public Node getNext()
   {
      return next;
   }

   public void setNext(Node n)
   {
      next = n;
   }

   Node(Object newItem, Node nextNode)
   {
      item = newItem;
      next = nextNode;
   }
}
```

The following gets you started on a linked-list class that uses it:

```
public class ListReferenceBased
{
   private Node head;
   private int size;

   public getHead()
   {
       return head;
   }
```

```
    private setHead(Node h)
    {
        head = h;
    }

    public getSize()
    {
        return size;
    }

    private setSize(int n)
    {
        size = n;
    }
}
```

1. Add the following method to the class. Assume that your list has no dummy node at the front of the list. Because of the precondition, you don't have to perform error handling. Notice that the positions are numbered starting at 1, not at 0.

```
// Locates a specified node in a linked list
// Precondition:  1 <= pos <= getSize()
// Postcondition:  returns a reference to the node at index i
private Node find(int pos)
```

Solution:

```
private Node find(int pos)
{
   Node curr = getHead();
   for (int i = 1; i <= pos; i++)
      curr = curr.getNext();
   return curr;
}
```

2. Add the following to the set of methods for this class. Because of the time constraint on the exam and the (artificial) precondition, you are not required to do any error handling. You may call `find`, above. **Don't forget to update the recorded size of the list.**

```
// precondition:  1 <= pos <= getSize() + 1
// postcondition:  item has been inserted at position i
public void add(int pos, Object item)
```

Solution:

```
public void add(int pos, Object item)
{
   if (pos == 1)
   {
      Node N = new Node (item, getHead());
      setHead(N);
   }
   else
   {
      Node Prev = find(pos-1);
      Node N = new Node (item, Prev.getNext());
      Prev.setNext(N);
   }
   setSize(getSize() + 1);
}
```

3. Add the following to the set of methods for this class. Proceed as in the previous problem.

```
// precondition:  1 <= pos <= getSize()
// postcondition:  the element at position i has been removed and returned
public Object remove (int pos)
```

Solution:

```
public Object remove(int pos)
{
    Node N;
    if (pos == 1)
    {
        N = getHead();
        setHead(getHead().getNext());
    }
    else
    {
        Node Prev = find(pos-1);
        N = Prev.getNext();
        Prev.setNext(N.getNext());
    }
    N.setNext(null);
    setSize(getSize() - 1);
    return N.getItem();
}
```

4. When $r \neq 1$, we often have to find $\sum_{i=0}^{n} r^i$. I claim that the righthand side of the following gives a nice shortcut: $\sum_{i=0}^{n} r^i = r^0 + r^1 + \ldots + r^n = (r^{n+1} - 1)/(r - 1)$ for all $r \neq 1$ **and for all** $n \geq 0$.

   (a) Prove the base case. (Don't forget that I've claimed that the identity holds for all $n \geq 0$.)

   Solution: $\sum_{i=0}^{0} r^i = r^0 = 1$; $(r^{0+1} - 1)/(r - 1) = (r - 1)/(r - 1) = 1$.

   (b) Using an equality involving a non-negative integer k, state the induction hypothesis.

   Solution: $\sum_{i=0}^{k} r^i = (r^{k+1} - 1)/(r - 1)$

   (c) Using an equality involving this k, state what you need to show in the induction step.

   Solution: $\sum_{i=0}^{k+1} r^i = (r^{k+2} - 1)/(r - 1)$

   (d) Using algebra, carry out the induction step.

   Solution: $\sum_{i=0}^{k+1} r^i = \sum_{i=1}^{k} r^i + r^{k+1} = (r^{k+1} - 1)/(r - 1) + r^{k+1}$. Putting both terms over the same denominator, we get that this is $(r^{k+1} - 1 + r^{k+2} - r^{k+1})/(r - 1)$. Simplifying, we get that this is $(r^{k+2} - 1)/(r - 1)$.

5. Counting.

(a) How many ways are there to select a string of eight alphanumeric characters if the string must have **exactly one digit**? (An alphanumeric character is a lower-case letter or a digit.)

Solution: *Eight choices for the position of the digit, ten ways to pick the digit,* 26^7 *ways to pick the letters for the five remaining five positions.* $8 * 10 * 26^7$.

(b) How many ways are there to select a string of eight alphanumeric characters if the string must have **at least** one digit?

Solution: *(Similar to a homework problem.)* 36^8 *good and bad strings of length 8.* 26^8 *bad strings of length 8.* $36^8 - 26^8$ *good strings of length 8.*

(c) How many ways are there to select a string of eight alphanumeric characters if the string must have **exactly three** digits **and no letter or digit can be repeated**?

Solution: $C(8, 3)$ *ways to pick the positions of the digits.* $10!/7!$ *ways to pick the digits to go into those positions.* $26!/21!$ *ways to pick the letters to go into the remaining positions.* $C(8, 3)(10!/7!)(26!/21!)$

6. How many bit strings of length 10 have a 0 in the first position or a 0 in the second position (or both). *Hint: Count how many have a 0 in the first position, and how many have a zero in the second position. Then figure out how many have been counted twice.*

   Solution: *(Similar to a problem in the assigned readings about the number of bit strings that begin or end with a 1.) Of the bit strings of length 10,* 2^9 *have a zero in the first position and* 2^9 *have a zero in the second position. We've counted the ones that have a zero in both positions twice. There are* 2^8 *of these. Subtracting the double-counted strings, we get* $2 * 2^9 - 2^8 = 2^{10} - 2^8$.

7. Let n be any positive integer.

   (a) Show that in any set S of $n + 1$ positive integers, there are two elements of S that have the same remainder when divided by n. (State what the pigeons are and what the holes are, and which hole a pigeon goes in. Tell how many pigeons and holes there are, and why this implies the answer.)

   Solution: *Pigeons: the $n + 1$ numbers in the set. A pigeon goes in hole i if its remainder is i when divided by n. There are n holes and $n + 1$ pigeons, so two pigeons end up in the same hole. That is, two of the numbers have the same remainder when divided by n.*

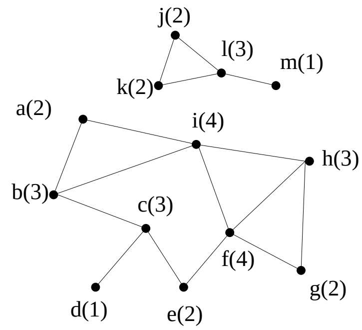
   (b) Use this to show that the difference between some pair of elements of S is a multiple of n.

   Solution: *If two of them have the same remainder when divided by n, their difference is divisible by n.*

   (c) Show that there is some multiple of n whose representation in base ten consists only of zeros and ones.

   Solution: *(Taken from the assigned readings.) Consider the first $n+1$ numbers in the sequence $(1, 11, 111, 1111, \ldots)$. The difference of two of these is divisible by n, by what we showed above, and the digits of this difference are all 1's and 0's.*

8. The *degree* of a vertex in an undirected graph is the number of edges that go out of the vertex. In the following graph, the degrees of the vertices are listed next to their names:



The ones that have odd degrees are $\{b, c, d, h, l, m\}$. The remaining vertices all have even degree. *Notice that there is an even number of vertices with odd degree; there are six of them.*

Show by induction on the number of edges that every undirected graph has an even number of odd-degree vertices:

(a) For the base case, explain why it is true for every graph with no edges.

Solution: *All vertices have degree 0, so there are 0 odd-degree vertices, which is an even number of them.*

(b) As your induction step, suppose $k \geq 0$ and that every graph with k edges has an even number of odd-degree vertices. Let G be an arbitrary undirected graph with $k+1$ edges. Since G is arbitrary, if you can show that G has an even number of odd-degree vertices, then this must be true for all graphs with $k+1$ vertices.

Show that G has an even number of odd-degree vertices by removing one edge to obtain a graph G' with k edges. Use the induction hypothesis to argue that G' has an even number of odd-degree vertices. Consider three cases in arguing that this must also be the case for G.

Solution:*Let u and v be the end vertices of an edge. Remove the edge to get G'. If u and v have odd degrees in G, they have even degrees in G'. so G has two more odd-degree vertices than G' does. If they have even degrees in G, they have odd degrees in G', so G has two fewer odd-degree vertices than G' does. If one of u and one of v has even degree and one has odd degree in G, then this is true in G' also, so G and G' have the same number of even and odd degree vertices.*

By the induction hypothesis, G' has an even number of odd-degree vertices. G either has two more, two fewer, or the same number of odd-degree vertices, so it also has an even number of them.

9. Prove that $\sum_{i=0}^{n}(C(n,i)C(n,n-i)) = C(2n,n)$. *Hint: How many ways are there to select* n *people from a set that consists of* n *men and* n *women?*

Solution: *The* n *men and* n *women are a set of size* $2n$*, so there are* $C(2n,n)$ *ways to select* n *people from them.*

When you select the n *people, you select 0 women and* n *men, or 1 woman and* $n-1$ *men, or 2 women and* $n-2$ *men, etc. up through* n *women and 0 men. The number of ways to do it the first way is* $C(n,0)C(n,n)$ *by the product rule. The number of ways to do it the second way is* $C(n,1)C(n,n-1)$*, etc.*

By the sum rule, the number of ways altogether is $\sum_{i=0}^{n} C(n,i)C(n,n-i)$

Therefore, $\sum_{i=0}^{n}(C(n,i)C(n,n-i))$ *and* $C(2n,n)$ *both count the number of ways to select the group of* n *people from a set of* n *men and* n *women, so they must be equal.*

10. Show that among any $n+1$ positive integers not exceeding $2n$ there must be an integer that divides one of the other integers.

    Solution: *(Taken from the assigned readings.) For each of integers i in the set, keep dividing it by two until you get an odd number i_o. Now i is a pigeon and i_o is its hole. There are n holes and $n+1$ pigeons, so two of them landed in the same hole. One of these must be a power of two times the other.*