COL100: Lab 9
Recursion

March 13, 2017

Part 1: Questions

1. Write a recursive code to find the sum of all elements of a list.

2. Write a recursive program to count the number of digits in a number. **Hint :** Repeated division by 10. Modify the above code to reverse a number. eg: 123 to 321

3. Fibonacci numbers are defined as follows:
   F(0) = F(1) = 1
   F(n) = F(n-1) + F(n-2)
   Write a recursive code to compute the nth fibonacci number. Use your program to compute the 10th, 20th, 30th and 40th Fibonacci numbers. Why does it take so much longer to computer the higher Fibonacci numbers? Can you make it better. **Hint :** Memoization (Read about this)

4. Write a recursive program to find the gcd of two numbers. **Hint :** Read about euclidean algorithm and comment on the recursion depth. Here is the iterative version:

```
function gcd(a, b)
    while b is not 0
       t := b;
       b := a mod b;
       a := t;
    return a;
```

5. Following is the code to find the kth power of n : (k is non-negative)

```
def pow(n, k):
    if(k == 0):
        return 1
    else
        return n * pow(n, k - 1)
```

   The recursion depth of this is k. Can you write a program with recursion depth log(k)? **Hint:** Think in terms of binary representation of k.

Part 2: Practice Questions

1. Write the recursive code to print the binary representation decimal number (base 10 number).

2. Following is the template for recursive binary search, complete it:

```
def bSearch(l, lo, hi, item): # l of type list, lo, hi, item of type int
  if(lo >= hi):
    print "NOT FOUND"
    return
  mid = lo + (hi - lo)/2
  if(l[mid] == item):
    print "found"
    return
  if(l[mid] > item):
    # complete this
  if(l[mid] < item):
    # complete this
```

3. Given two sorted Lists l1 and l2, write a program to merge these two into l3 such that l3 is sorted. Eg l1 = [1,2,4], l2 = [3,5,7] Then l3 = [1,2,3,4,5,7]. Following is the iterative code:

```
def merge(l1, l2):
  sz1 = len(l1)
  sz2 = len(l2)
  i = 0
  j = 0
  k = 0
  l3 = []
  while(i < sz1 and j < sz2):
        if(l1[i] >= l2[j]):
            l3[k] = l2[j]
            k += 1
            j += 1
        else:
          l3[k] = l1[i]
            k += 1
            i += 1
    while(i < sz1):
      l3[k] = l1[i]
        k += 1
        i += 1

  while(j < sz2):
    l3[k] = l2[j]
        k += 1
        j += 1
  return l3
```

Write the recursive code that achieves the same.

4. You have a phone book stored as a list of tuples, sorted by name. eg [('aon', 123), ('bob', 234) ...]. Given a name find all the numbers corresponding to a name. Write a recursive binary search based code to do this.

Part 3: Challenge

1. Fractals are recursive drawings. They occur a lot in nature and there is a field called fractal geometry. Can use recursion to draw them.

2. Given an integer K, Print All binary string of size K without consecutive ones. Examples: Input : K = 3, Output : 000 , 001 , 010 , 100 , 101. Input : K = 4, Output :0000 0001 0010 0100 0101 1000 1001 1010.

3. Given following three values, the task is to find the total number of maximum chocolates you can eat. **Money** : Money you have to buy chocolates, **Price** : Price of a chocolate, **Wrap** : Number of wrappers to

be returned for getting one extra chocolate. It may be assumed that all given values are positive integers and greater than 1. Examples:

(a) Input : money = 16, price = 2, wrap = 2, Output : 15. Price of a chocolate is 2. You can buy 8 chocolates from amount 16. You can return 8 wrappers back and get 4 more chocolates. Then you can return 4 wrappers and get 2 more chocolates. Finally you can return 2 wrappers to get 1 more chocolate.

(b) Input : money = 15, price = 1, wrap = 3, Output : 22. We buy and eat 15 chocolates. We return 15 wrappers and get 5 more chocolates. We return 3 wrappers, get 1 chocolate and eat it (keep 2 wrappers). Now we have 3 wrappers. return 3 and get 1 more chocolate. So total chocolates = 15 + 5 + 1 + 1

(c) Input : money = 20, price = 3, wrap = 5, Output : 7