CSC373 Week 4 Tutorial

November 8, 2017

1 Fibonacci Sequence

How to calculate the n'th value in the Fibonacci Sequence?
Recursive version:

```
1  def fib1(n):
2          if n<=1:
3                  return n
4          return fib1(n-2) + fib1(n-1)
```

Trace the recursive calls of $fib1(6)$.

$$fib1(6) = fib1(5) + fib1(4)$$
$$= 2 * fib1(4) + fib1(3)$$
$$= 3 * fib(3) + 2 * fib(2)$$

... There are many duplicate fib1 calls which wastes time/power. How can we avoid recalculating information?
Dynamic Programming!
Create bottom-up approach instead of top-down to save previous fib1 calls.

```
1  def fib2(n):
2  memo = [0,1]
3  while n < len(memo):
4          memo.append(memo[-1]+memo[-2])
5  return memo[n]
```

2 Binary strings - No consecutive 1s

Given, n, find the number of binary strings of size n that have no consecutive 1s.
Let $a[i]$ represent the number of binary strings of length i that end with a 0.
Let $b[i]$ represent the number of binary strings of length i that end with a 1.

$$a[i] = a[i-1] + b[i-1]$$
$$b[i] = a[i-1]$$