Lecture 12 Solving First Order Recurrence Relations

Grimaldi Chapter 10.1

$p_n = n\, p_{n-1}$

$c_n = (n-1) + c_{n-1}$

$h_n = 1 + 2h_{n-1}$

```
1: void Bubblesort( double A[], int n ) {
2: // sort the array A of size n into ascending order
3:     int i; double t;
4:     if( n==1 ) return;
5:     for( i=1; i<=n-1; i++ )
6:         if( A[i-1] > A[i] ) {
7:              t = A[i-1]; A[i-1] = A[i]; A[i] = t;
8:         }
9:     Bubblesort(A,n-1);
10:    return;
11: }
```

What should we count to determine the cost of the Bubblesort algorithm?
We will count the number of comparisons between elements of A in line 6.
Let c_n be the number of comparisons.

A First Look at Solving Recurrence Relations.

How can we solve RRs like

(1) $b_n = 2b_{n-1}$ for $n \geq 2$ and $b_1 = 2$.

(1) $c_n = c_{n-1} + (n-1)$ for $n \geq 2$ and $c_1 = 0$.

Example $b_n = 2b_{n-1}$

Example $c_n = c_{n-1} + (n-1)$

Theorem (A different way to solve first order RRs)

Every sequence $x_0, x_1, x_2, \ldots$ *satisfying the recurrence* $x_n = dx_{n-1}$ *has the general solution* $x_n = cd^n$ *for some constant* c*. (The sequence is a geometric progression.)*

Proof (substitution)

This suggests the following general strategy for solving RRs:

(1) Find the **general solution** to the RR. This will have one or more constants. Note: a RR of order k will has k constants.

(2) Use the k initial values to determine the constants. This gives a **unique** solution.

Example. Suppose $x_n = 5x_{n-1}$ and $x_0 = 7$. First find the general solution then the unique solution satisfying $x_0 = 7$.

Exercise 1. Solve $p_n = np_{n-1}$ where $p_1 = 1$.

Exercise 2. Solve $x_n = x_{n-1} + An + B$ for $x_1 = C$.