Recursion examples

Example The function $f(n)=2^n$, where n is a natural number, can be defined recursively as follows:

1 Initial Condition: $f(0) = 1$,
2 Recursion: $f(n+1) = 2 \cdot f(n)$, for $n \geq 0$.

Here is how the definition gives us the first few powers of 2:
$2^1 = 2^{0+1} = 2^0 \cdot 2 = 2$
$2^2 = 2^{1+1} = 2^1 \cdot 2 = 2 \cdot 2 = 4$
$2^3 = 2^{2+1} = 2^2 \cdot 2 = 4 \cdot 2 = 8$

Recursive Pseudocode: FUNCTION 2POWER(N: INTEGER) RETURNS INTEGER

Iterative Pseudocode: FUNCTION 2POWER(N: INTEGER) RETURNS INTEGER

Example: Recursive Algorithm for Sequential Search

Algorithm SeqSearch(*L, i, j, x*)
Input: *L* is an array, *i* and *j* are positive integers, *i* ≤ *j*, and *x* is the key to be searched for in *L*.
Output: If *x* is in *L* between indexes *i* and *j*, then output its index, else output *0*.

```
Algorithm:
   IF  i ≤ j  THEN
        IF  L(i) = x, THEN
              return i ;
        ELSE
              return SeqSearch(L, i+1, j, x)
       ENDIF
   ELSE
        return 0.
   ENDIF
END
```

Recursive Pseudocode for the above algorithm.

Example: greatest common divisor

Therefore to write the recursive program for finding the greatest common divisor of any two numbers, we have to express the greatest common divisor (GCD) function in a recursive form:

1. if m <= n and n%m = 0 then gcd(n,m) = m (termination step).
2. if n < m then gcd(n,m) = gcd(m,n) (recursive definition of greatest common divisor).
3. if n >= m then gcd(n,m) = gcd(m,n%m) (recursive definition of greatest common divisor).

```
gcd(n, m)
       IF m<=n AND n%m = 0 THEN     *********Rematk: n % m means reminder of n divided by m
           return m
       ENDIF
       IF n < m THEN
         return gcd(m, n)
       ELSE
         return gcd(m, n%m)
       ENDIF
END
```

Recursive Pseudocode for the above algorithm.