James Keller

ITCS 2215 - 02

Project 1

**Exercises from 2.4**

**5. *Towers of Hanoi***:

A. 2^n - 1 is the total number of moves needed per n (number of disks). 2^64 - 1 = 1.8446744073709551615 x 10^19 divided by the total number of minutes in a year (525949) = 3.50732562 x 10^13 years

B. The number of moves of the ith disk, defined by 0< = ith <= n, is 2^(i-1) where i is equal to the number of the disk (disks are counted in ascending order from largest to smallest).

Example: The total number of disks is 3. The largest disk is 1, the medium disk is 2, and the smallest disk is 3. Therefore, 2^(1-1) = 1, 2^(2-1) = 2, and 2^(3-1) = 4.

C. Obtain the number of disks and determine the number of iterations using a bit shift instead of a power function. Incorporate that into a for loop. Call the disk method to return the disk to be moved. Source is the tower currently occupied by the disk and the destination is the target tower. The source is calculated by direction (1 = clockwise, 2 = counterclockwise) times movement (times d is moved in stage i) mod 3. The target is calculated is calculating by summing the source and direction mod 3. Finally print the output.

*n = number of disks*

*limit = (1 << n) - 1;*

*for (i = 0; i < limit; i++)*

*d = disk(i);*

*source = (movements()\*direction()%3;*

*target = (source + direction)%3;*

*Print ouput*

*disk()*

*x = i+1*

*for (j = 0; x%2 == 0; j++)*

*x /= 2*

*return j*

*movements()*

*return ((i >> d) + 1) >> 1*

*direction()*

*return 2 - (n + d)%2*

**6. Restricted *Towers of Hanoi***:

Algorithm RTH (numDisks)

//Input number of disks requested for the puzzle

//Output Disk moves to complete the puzzle

//(n-1) = a subset of the total disks excluding the nth disk

//nth disk = largest disk

if n = 1

move n disk from source to spare

move n disk from spare to target

else

move RTH(n-1) disks from source to spare

move RTH(n-1) disks from spare to target

move nth disk from source to spare

move RTH(n-1) disks from target to spare

move RTH(n-1) disks from spare to source

move nth disk from spare to target

move RTH(n-1) disks from source to spare

move RTH(n-1) disks from spare to target

The steps mentioned above could be reduced if it were understood that the RTH(n-1) moves were made by way of the spare. In other words moving RTH(n-1) disks from the source to the target are always accomplished using the spare peg. source 🡪 spare 🡪 target

if n = 1

move n disk from source to target

else

move RTH(n-1) disks from source to target

move nth disk from source to spare

move RTH(n-1) disks from target to source

move nth disk from spare to target

move RTH(n-1) disks from source to target

Simple cases result in x(1) = 1 and x(2) = 8: x(n) = 3^n-1 giving a time complexity of theta(3^n). Total moves = 3^n-1 where n is the total number of disks.