**Assignment - 3**

**2**.  Give a recursive algorithm to compute the product of two positive integers, m and n, using only addition and subtraction.

Input:- 2 integers m & n

Output:- to compute the product of two positive integers

If n = > 1 then

Return m

Else

Return (m + (m x (n-1))

**4**. Order the following functions by asymptotic growth rate.

4nlogn + 2n, 210, 2log n, 3n + 100logn, 4n, 2n, n2 + 10n, n3, nlogn

Solution: 210 , 2 log n, 3n + 100logn , 4n , 4nlogn + 2n , nlogn , n2 + 10n , n3, 2n

**5**. Suppose you are given an n-element array containing distinct integers that are listed in increasing order. Given a number k, describe a recursive algorithm to find two integers in a that sim to k, if suck a pair exists. What is the running time of your algorithm?

Input: First, start with the leftmost index: I = 0

Second, create the rightmost index: r = arraySize - 1

Output: two integers in A that sum to k, if such a pair exists

While (I < r)

If(A[i] + A[r] == sum) then

Return 1

Else if (A[i] + A[r] < sum) then

Return 1

Else (r--)

return 0

c

the running time of algorithm:

O(n^2)

**6**. An evil king has n bottles of wines, and a spy has just poisoned one of them.  Unfortunately, they do not know which one it is.  The poison is very deadly; just one drop diluted even a billion to one will still kill.  Even so, it takes a full month for the poison to take effect.  Design a scheme for determining exactly which one of the wine bottles was poisoned in just one month’s time while expending O(log n) taste testers.

Tasters to cups, cups to bits, tasters to bits, tasters to bits, tasters to bits, tasters to bits, tasters to bits, tasters to bits, tasters to bits, tasters to bits, tasters Some of your tasters will be deceased in a month.

Tasters 3 and 1 die, for example.

Set all other bits to 0 and the corresponding bits to 1. The poisoned bottle will be identified by the binary number 00000101 that results.

the number of people who have tried the wines f(n) = (int)(logn)+1 , so f(n) is O(logn)