**Assignment Day 4 | 28th December 2020**

**Question-1:**

In the Binary Search algorithm, it is suggested to calculate the mid as beg + (end - beg) / 2 instead of (beg + end) / 2. Why is it so?

**Answer-1:**

In the Binary search ‘(beg+end)/2’ is subject to overflow with large values. With iteration it’s not valid.

In ‘beg + (end - beg) / 2’ avoids the overflow and this also works with iterators, as the outcome for (end-beg) is a number. As we can subtract two iterators to get the distance between them, but we can’t add two iterators.

For example:

beg= m\_int-3 & end= m\_int-1 then

beg+end would be larger than m\_int, but end-beg, would just be 2.

**Question-2:**

Write the algorithm/function for Ternary Search.

**Answer-2:**

int ternary\_search (int beg, int end, int key, int ar [])

{

if(end>=1)

{

int mid1=beg + (end-beg)/3;

int mid2 = end – (end-beg)/3;

if(ar[mid1] ==key)

{

return mid1;

}

if(ar[mid2] ==key)

{

return mid2;

}

if(key<ar[mid1])

{

return ternary\_search (beg, mid1-1, key, ar);

}

elseif(key>ar[mid2])

{

return ternary\_search (mid2+1, end, key, ar);

}

else

{

return ternary\_search (mid1+1, mid2-1, key, ar);

}

}

return -1;

}