Assignment Day 4

Raj Sandip Kaste

Q1. 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?

Ans: Using mid = (beg + end) / 2 sets mid to the average of beg and end, truncated down to the nearest integer. On the face of it, this assertion might appear correct, but it fails for large values of the int variables low and high. Specifically, it fails if the sum of low and high is greater than the maximum positive int value (2^(31) - 1).

The sum overflows to a negative value, and the value stays negative when divided by two. In C this causes an array index out of bounds with unpredictable results. This bug can manifest itself for arrays whose length (in elements) is 230 or greater (roughly a billion elements).

Hence it is suggested to calculate the mid as beg + (end - beg) / 2 instead of (beg + end) / 2.

Q2. Write the algorithm/function for Ternary Search.

```
Ans:
int ternary_search(int l,int r, int x)
{
    if(r>=l)
    {
        int mid1 = l + (r-l)/3;
        int mid2 = r -  (r-l)/3;
        if(ar[mid1] == x)
            return mid1;
        if(ar[mid2] == x)
            return mid2;
        if(x<ar[mid1])
            return ternary_search(l,mid1-1,x);
        else if(x>ar[mid2])
            return ternary_search(mid2+1,r,x);
        else
            return ternary_search(mid1+1,mid2-1,x);

    }
```

```
    return -1;
}
```