

THIS IS AN ALGORITHM WHICH I HAVE DEVELOPED AND NAMED AS "MY -SEARCH". IT WILL SEARCH A REQUIRED ELEMENT FROM THE ARRAY.THE ALGORITHM IS LIKE THIS :-

ALGO-MY SEARCH

```
VAR:A[ ],N,I,J,C,ITEM;
I=1,J=N,C=0;
WHILE((I<N/2) || (J>=N/2))
{
    IF(A[I]==ITEM)
    {
        WRITE"ELEMENT FOUND";
        C++;
        EXIT();
    }
    ELSE IF(A[J]==ITEM)
    {
        WRITE"ELEMENT FOUND";
        C++;
        EXIT();
    }
    I++;
    J--;
}
IF(C==0)
{
    WRITE"ELEMENT NOT FOUND";
}
```

THIS IS A ALGORITHM WHICH HAS THE SAME TIME COMPLEXITY AS THAT OF LINEAR SEARCH OF $O(n)$. BUT STILL IT IS BETTER THAN LINEAR SEARCH IN TERMS OF EXECUTION TIME. LET $A[]$ BE THE THE ARRAY OF SOME SIZE N . IF THE ELEMENT WHICH WE WANT TO SEARCH IS AT ANY POSITION BEFORE $N/2$ THAN MY-SEARCH AND LINEAR-SEARCH BOTH WILL HAVE EXECUTION TIME , BUT THE MAGIC HAAPENS WHEN THE SEARCH ELEMENT IS AFTER $N/2$ POSITION.SUPPOSE THE ELEMENT WANT TO SEARCH IS AT N th POSITION, THEN USING THE LINEAR SEARCH WILL FIND THE ELEMENT AFTER N th ITERATION,BUT USING MY-SEARCH WE CAN SEARCH THE ELEMENT AFTER 1st ITERATION ITSELF.

WHEN WE ARE DEALING WITH A SITUTATION WHEN SIZE IS SOMETHING 10 OR 15 ITS OK. BUT CAN YOU IMAGINE THE CASE WHEN THE SIZE IS "100000000" OR EQUIVALANENT.IF WE USE THIS LINEAR SEARCH TECHINIUQE THAN THE TOTAL EXPENDITURE YOU CAN THINK OFF TO CONTINUE THE LOOP FOR 100000000 TIMES.BUT RATHER IF USE MY-SEARCH U GET THE DESIRED SEARCH JUST AFTER 1 ITERATIONS.

SO ,NOW YOU CAN IMAGINE HOW WE CAN PREVENT SUCH A BIG LOSS THROUGH MY-SEARCH.

THANK YOU