

## HW 03: Exponential Search (Concept)

**Exponential search involves two steps:**

**Step 01:** Find range where element is present

**Step 02:** Do Binary Search in above found range.

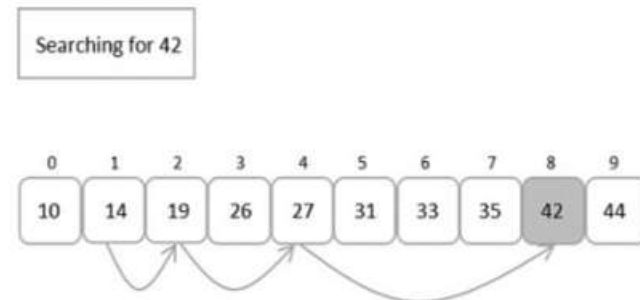
**Exponential search applications:**

Search in  $\infty$  infinite array (Sorted array)

Better than binary search when target element is near in beginning

**Time complexity:**  $O(\log n)$

**Space complexity:**  $O(1)$



$x =$ 

13
----

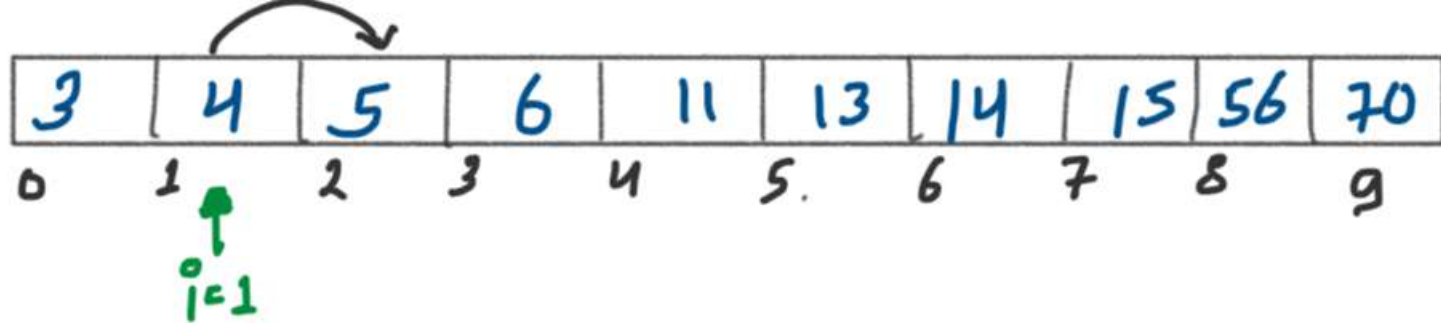
3	4	5	6	11	13	14	15	56	70
0	1	2	3	4	5	6	7	8	9

Step 01: Find range where element is present

3	4	5	6	11	13	14	15	56	70
0	1	2	3	4	5	6	7	8	9

$(arr[0] == x)$   
↪ return 0

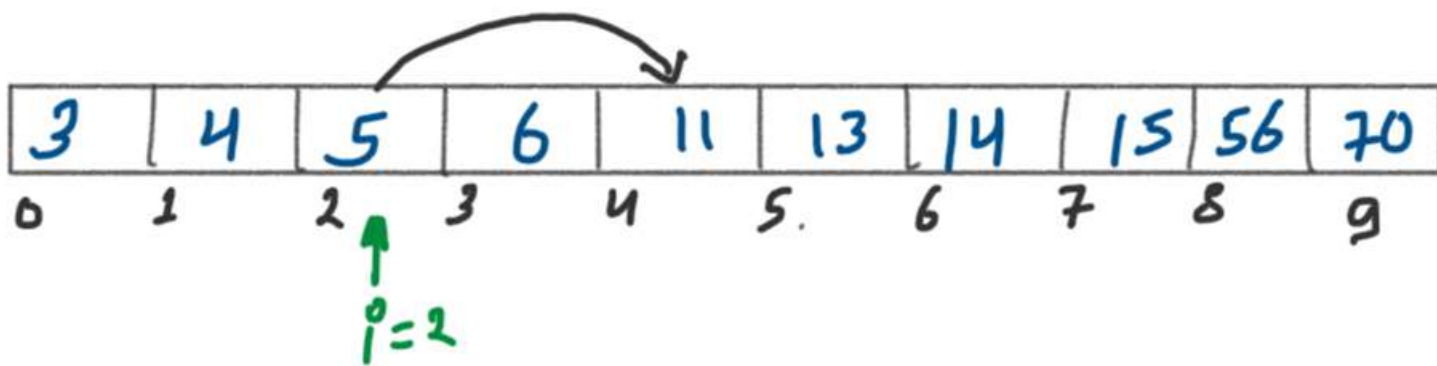
Iteration: 1



$arr[i] \leq X$   
 $\rightarrow i = i * 2$

$4 < 13$

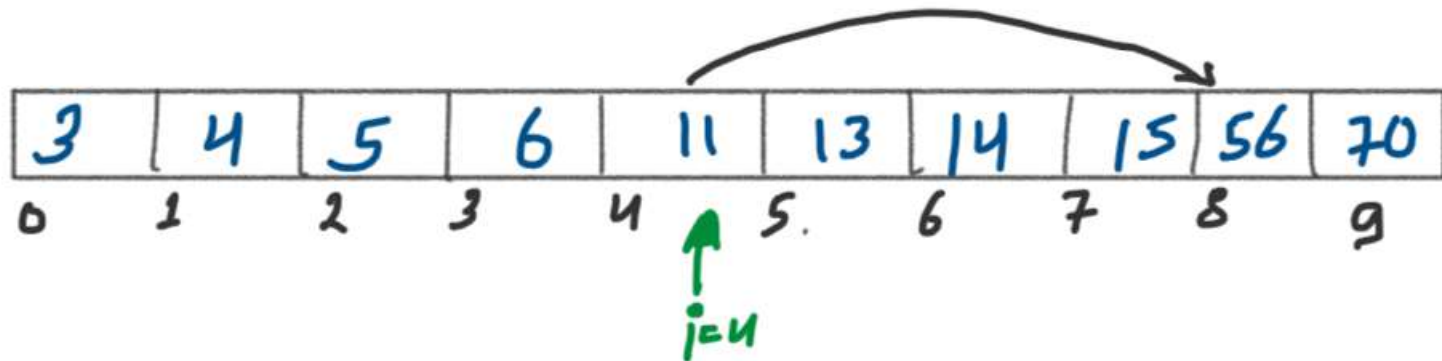
Iteration: 2



$arr[i] \leq X$   
 $\rightarrow i = i * 2$

$5 < 13$

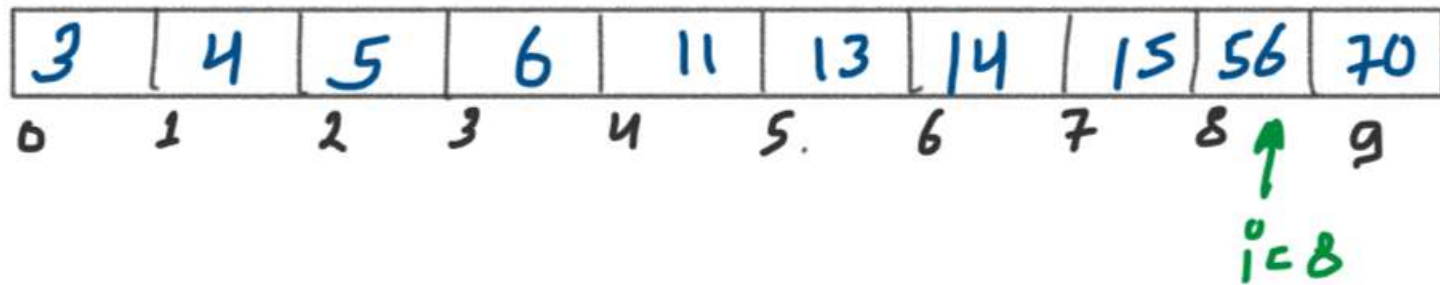
Iteration: 3



$arr[i] <= X$   
 $\rightarrow i = i * 2$

$11 < 13$

Iteration: 4



Return  
 $\rightarrow \text{BS}(\text{arr}, \frac{i}{2}, \min(i, n-1), x)$   $56 < 13$  **Fail** } Ruk jao... ---

Start  $\swarrow$   $\searrow$  END  $\searrow$  Target

**Range: [4, 8]**

$\uparrow$   
 val at index 5

Step 02: Do Binary Search in above found range.

11	13	14	15	56
4	5	6	7	8



```

// HW 03: Exponential Search (Concept)
#include<iostream>
#include<vector>
using namespace std;

// Step 02: Do Binary Search in above found range
int binarySearch(vector<int>& arr, int start, int end, int target){
    int mid = start + (end - start)/2;
    while(start<=end){
        if(arr[mid]==target){
            return mid;
        }
        else if(arr[mid]<target){
            start = mid + 1;
        }
        else{
            end = mid - 1;
        }
        mid = start + (end - start)/2;
    }
    return -1;
}

// Step 01: Find range where element is present
int exponentialSearch(vector<int>& arr,int x){
    int n = arr.size();

    if(arr[0]==x){
        return 0;
    }

    int i = 1;
    while(i<n && arr[i]<=x){
        i = i*2; // i = i<<1 or i*=2
    }
    return binarySearch(arr, i/2, min(i,n-1), x);
}

// Main method
int main(){
    vector<int> arr{3,4,5,6,11,13,14,15,56,70};
    int target;
    cin>>target;

    int ans = exponentialSearch(arr,target);
    cout<<"Element "<<target<<" found at index "<<ans<<endl;
    return 0;
}

/*
Given a sorted array, and an element x to be searched, find position of x in the array.
Examples 01:
Input: arr[] = {10, 20, 40, 45, 55} and x = 45
Output: Element 45 found at index 3

Examples 02:
Input: arr[] = {10, 15, 25, 45, 55} and x = 15
Output: Element 15 found at index 1
*/

```