```
Algorithm BinarySearch(array, first, last):

    if (first < last) :

        mid = (first + last) / 2
        if (-1 < mid - 1 && array[mid - 1] <= array[mid] && mid + 1 < array.length && array[mid + 1] <= array[mid]) :
            return array[mid]

        if (-1 < mid - 1 && mid + 1 < array.length && array[mid - 1] > array[mid + 1]) :
            last = mid - 1
        else if (mid + 1 < array.length ) :
            first = mid + 1


    if (array[0] < array[array.length - 1]):
        return array[array.length - 1]
    else:
        return array[0]



n = input()
arr[n]

for (i = 0; i < n; i++):
   arr[i] = input ()


print(BinarySearch(arr, 0, n - 1))
```

پیچیدگی زمانی BinarySearch() :

T(n) = T(n/2) + f (n), O(f(n)) < log_2^n
T(n) $\in \theta(log_2^n)$