Homework-4

Ali Valiyev

November 2021

1 Homework solution

Task 1

Question: Suppose you are given a sequence (i.e., array) of size n with each entry holding a distinct number. For some value p between 0 and $n-1$, the values in the array entries increase up to position p and then decrease on the remainder of the way until position $n-1$. An example of such array would be [12,17,38,54,55,69,68,44,39,19,14,7] where p and n could be 5 and 12, respectively. Develop an algorithm to find the peak entry p without having to read the entire array. The algorithm should read as few entries of the array as possible. Also note that p could be 0 or $n-1$, so your algorithm should be able to handle these corner cases as well.
Solution:

```python
def peak(arr, left, right):
    if left == right:
        return left

    if arr[left] < arr[right] and right == left + 1:
        return right

    middle = (left + right)//2

    if arr[left] >= arr[right] and right == left + 1:
        return left

    if arr[middle] > arr[middle + 1] and arr[middle] < arr[middle - 1]:
        return peak(arr, left, middle-1)

    if arr[middle] > arr[middle + 1] and arr[middle] > arr[middle - 1]:
        return middle

    else:
        return peak(arr, middle + 1, right)

arr = [12,17,38,54,55,69,68,44,39,19,14,7]
print ("The index of a peak is %d"% peak(arr, 0, len(arr)-1))
```

```
The index of a peak is 5
```

Total cost = T(n) = 2T(n/2) + n $\leq$ 2c*n/2Log(n/2) + n = c*nLog(n) - c*nLog(2)+n = c*nLog(n) - c*n + n $\leq$ c*nLog(n)
So, the growth-rate function for this algorithm is $O(nLog(n))$.