ECE374 SP23 HW4

Contributors

Zhirong Chen (zhirong4)

Ziyuan Chen (ziyuanc3)

Problem 5

Given an arbitrary array $A[1:n]$, describe an algorithm to determine in $O(n)$ time whether A contains more than $\frac{n}{4}$ copies of any value. **Do not use hashing, or radix sort, or any other method that depends on the precise input values.**

Solution

Intuition

- In a *sorted* array, elements with the same value occupy a contiguous subarray. Such a subarray with length over $\frac{n}{4}$ must cover at least one of the array's three quarters.
- In other words, if there is a value that appears more than $\frac{n}{4}$ times, **it must be among the three quarters**. This also applies to unsorted arrays.

Algorithm

```
$\text{IfFrequentValueExists}(A, n)$
    for $i \leftarrow 1$ to $3$
        $\text{quarter} \leftarrow \text{QuickSelect}(A, \lfloor \frac{in}{4} \rfloor)$
        if $\text{Count}(A, \text{quarter}) > \frac{n}{4}$
            return True
    return False
```

Runtime Analysis

- $\text{QuickSelect}(A, k)$ (covered in lecture) finds the k^{th} smallest element of A in $O(n)$ time.
- $\text{Count}(A, \text{target})$ scans the *unsorted* A and counts the occurrences of target. Evidently, it runs in $O(n)$ time.
- In total, there are **6 calls** to $O(n)$ algorithms regardless of the array's length. Therefore, $\text{IfFrequentValueExists}(A, n)$ runs in $O(n)$ time.