York University EECS 3101Z January 30, 2023

Homework Assignment #3
Due: February 6, 2023 at 7:00 p.m.

1. Let C be a class of objects. Let $A[1..n]$ be an array of references to objects of class C. You can test whether two references refer to the same object (as with the == operator in Java). However, you cannot test whether one object of type C is greater than another, because there is no comparison operator defined for class C.

An object o is called a *majority element* of the array $A[1..n]$ (where $n > 0$) if strictly more than $\frac{n}{2}$ entries of A contain a reference to o. Recall that we looked at the Boyer-Moore algorithm for finding a majority element in an array in the first lecture. Here we look at a different approach to the same problem.

Consider the following DISCARD routine, which splits the array $A[1..n]$ (where n is even) into $\frac{n}{2}$ pairs. For each pair, if the two elements of the pair are identical, one copy of that element is placed in another array B; otherwise neither element is copied into B.

function DISCARD($A[1..n]$)
    precondition: n is a positive even number and for each i, $A[i]$ is a reference to an object of class C
    $j \leftarrow 0$
    for $i \leftarrow 1..\frac{n}{2}$ **do**
        if $A[2i-1] = A[2i]$ **then**
            $j \leftarrow j + 1$
            $B[j] \leftarrow A[2i]$
        end if
    end for
    return $B[1..j]$
    postcondition: for all o, *if* o is a majority element of $A[1..n]$
        then $j > 0$ and o is a majority element of $B[1..j]$
end function

Here is a brief argument that DISCARD satisfies its postconditions. Suppose n is even and o is a majority element of $A[1..n]$.
Let a be the number of i's such that $A[2i-1] = A[2i] = o$.
Let b be the number of i's such that $A[2i-1] = A[2i] \neq o$.
Let c be the number of i's such that $A[2i-1] \neq A[2i]$.
The number of o's in A is at most $2a + c$, so $2a + c > \frac{n}{2}$ (since there are more than $\frac{n}{2}$ o's in A). So, $2a + c > \frac{n}{2} = a + b + c \Rightarrow 2a > a + b \Rightarrow a > \frac{a+b}{2} \Rightarrow a > \frac{j}{2}$. There are a o's in $B[1..j]$, so o is a majority element of B, as required to satisfy the postconditions.

[2] **(a)** Give an example of an array $A[1..n]$ (where n is a positive even number) such that $A[1..n]$ does not have a majority element, but the array $B[1..j]$ computed by the DISCARD routine does have a majority element.

[2] **(b)** Prove that if n is odd and o is a majority element of $A[1..n]$, then either o is a majority element of $A[1..n-1]$ or $o = A[n]$.

[6] **(c)** Give an algorithm called FINDMAJORITY that returns a majority element of $A[1..n]$ if it exists, or returns "no majority element" otherwise. Your algorithm should run in $O(n)$ time. You should explain briefly why your algorithm is correct and why its worst-case running time is $O(n)$, but you need not give a full, formal proof of correctness. You may assume that checking the equality of two references can be done in constant time.

Hint: using the information above, design a divide-and-conquer algorithm for FINDMAJORITY that uses the DISCARD routine as a subroutine.