Algorithms Homework- week 9

Massimo Hong
Student id: 2020280082

November 18, 2021

1

The run time complexity of the algorithm is $log(n)$, given that $m = n$. We only need to consider the operations inside the while loops, as the others cost 1.

- The outer loop ($m > 0$): can be executed a maximum of $C < m$ times.
- The inner loop ($!(m\%2)$): can be run a maximum of $log(m = n)$ times.

We can conclude that an upper bound for the run time complexity is given by $Clog(n) = O(log(n))$, for $n > 0$. Of course, if $n = 0$, it will have a $O(1)$ cost.

2

$$2n^2 - 3n = O(n^2)$$

With $n \geq 0$ and $k = 1$:

$$\frac{f(n)}{g(n)} = \frac{2n^2 - 3n}{n^2} \leq \frac{2n^2}{n^2} = 2$$

$C = 2$

$$2n^2 - 3n \leq 2n^2 = O(n^2)$$

3

1. The input of the Majority Element problem is the same as the input of a Sorting problem: An array of elements.
2. We already know several algorithms to solve a sorting problem, i.e. Insertion sort, with a time complexity of $\theta(nlogn)$.
3. The output of the sorting problem is an ordered array A. The majority element (if it exists) is the element at the position of $\lfloor \frac{A.length}{2} \rfloor$.

4

```
procedure LINEAR SEARCH(A, v)
    for i = 1 to A.length do
        if A[i] == v then
            return i
        end if
    end for
    return NIL
end procedure
```

Loop Invariant: At the start of the $\mathbf{i} - th$ iteration, the sub array $A[1....i-1]$ does not contain v.
Proof:
Initialization
At $i = 1$ we have an empty sub array, so the condition is obviously satisfied.
Maintenance
Suppose the invariant is satified at the start of the $i - th$ iteration, we can have:

- $A[i] = v$, A[i....i-1] does not contain v, and we can go to termination
- $A[i] \neq v$, $A[1....i]$ does not contain v

Termination

- $A[i] = v$, $A[1....i-1]$ does not contain v, termination
- We got to $i = A.length + 1$ and didn't find v. $A[1.....A.length]$ does not have v and the procedure returns NIL, termination.