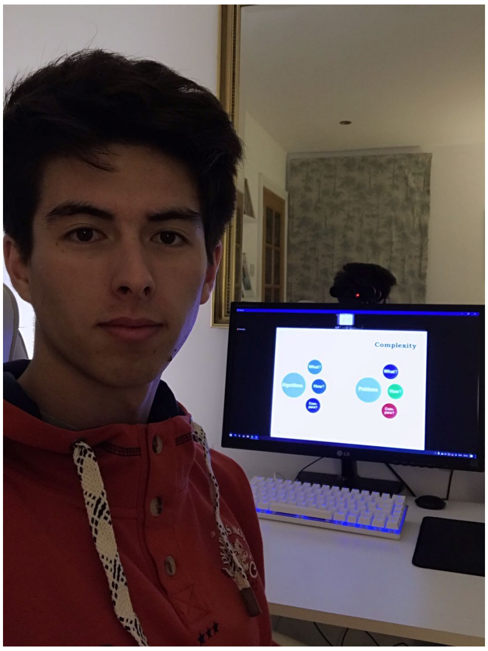
HWw9-1

Samuel Pegg 2020280261

November 2020

1



2

We seek positive constants c, n_0 such that $0 \leq 2n^2 - 3n \leq cn^2$ for all $n \geq n_0$. Since

$$\frac{d}{dn}\left(2n^2 - 3n\right) = 4n - 3 \geq 0 \;\forall\; n \in \mathbb{N} = \{1, 2, \dots\}$$

and

$$2n^2 - 3n = 0 \implies n = 0 \text{ or } n = \frac{3}{2}$$

we can conclude that $2n^2 - 3n \geq 0$ for all integers $n \geq 2$ since $2n^2 - 3n$ increases monotonically for $n > \frac{3}{2}$. Next, the inequality

$$2n^2 - 3n \leq cn^2 \iff 2 - \frac{3}{n} \leq c$$

is satisfied by $c = 2$ for all $n \in \mathbb{N}$. Hence $2n^2 - 3n = O(n^2)$ with $n_0 = 2$ and $c = 2$.

3

Let $k \geq d$. Again, we seek positive constants c, n_{min} such that $0 \leq p(n) \leq cn^k$ for all $n \geq n_{min}$. Since

$$p(n) = a_0 + a_1 n + \cdots + a_d n^d \leq |a_0| + |a_1|n + \cdots + |a_d|n^d \leq |a_0|n^d + |a_1|n^d + \cdots + |a_d|n^d$$

if we define

$$c := \sum_{i=0}^{d} |a_i| \geq 0$$

then we get

$$p(n) \leq cn^d \leq cn^k \quad \forall\, k \geq d$$

Similarly,

$$p(n) = a_0 + a_1 n + \cdots + a_d n^d \geq -|a_0| - |a_1|n - \cdots - |a_{d-1}|n^{d-1} + a_d n^d \geq (-|a_0| - |a_1| - \cdots - |a_{d-1}|)n^{d-1} + a_d n^d \geq 0$$

holds if we have

$$n \geq \frac{|a_0| + \cdots + |a_{d-1}|}{a_d} =: n_{min}$$

So $p(n) = O(n^k)$ for all $k \geq d$ with

$$c = \sum_{i=0}^{d} |a_i|$$

and

$$n_{min} = \frac{|a_0| + \cdots + |a_{d-1}|}{a_d}$$

4

```
candidate = NIL
count = 0
for i = 1 to n
        if count == 0
            candidate = A[i]
        if candidate == A[i]
            count = count + 1
        else count = count − 1
```

1. Transform the input: The input of the majority element algorithm is just a list A, which is the same as the input for a sorting problem. Thus no transformation is required.

2. Run the sorting algorithm: this produces a sorted list A'.

3. Transform the output: take the middle element of A'. Assume the index of the first element of A' is 1, that means taking the in the $\lceil \frac{n}{2} \rceil$ position. This will be the same as the candidate produced in the majority element algorithm.

Finally, we check that the output is indeed the majority element. If A contains a majority element, then the output will be correct.