BU CLA CS 530: Analysis of Algorithms

Spring 1996

Problem Set 7

Due Thursday 96.03.14, at 11:00 am, in class. Late submissions not accepted. Reminder: The mid-term exam is on Thursday 96.03.14. Plan your time accordingly, to do this assignment and prepare for the exam.

---

1. Given an array $A[1\,..\,n]$ of $n \geqslant 2$ (not 1) integers, an integer x is a *(simple) majority* in A if it occurs more than $n/2$ times in A. The following is an algorithm to find a majority element in A, if it exists. (The case $n = 1$ is trivial, because $A[1]$ is always a majority in this case.)

```
MAJORITY(A)
n ← length[A]
C ← A[1]
M ← 1
N ← 0
k ← 1
ℓ ← 1
▷ If M = 0 then there is no majority in A[1 .. ℓ]
▷ If M ≠ 0 then C is the majority in A[k .. ℓ]
▷ with multiplicity M + N, where M + 2N = ℓ − k + 1
repeat ℓ ← ℓ + 1
            if M = 0
            then C ← A[ℓ]
                 M ← 1
                 N ← 0
                 k ← ℓ
            else if C = A[ℓ]
                 then M ← M + 1
                 else M ← M − 1
                      N ← N + 1
until  ℓ = n
▷ Check whether C is the majority in A[1 .. n]
if M = 0
then return "there is no majority element in A"
else count ← 0
     for ℓ ← 1 to n do
         if A[ℓ] = C then count ← count + 1
     if count > n/2
     then return "C is a majority element in A"
     else return "there is no majority element in A"
```

The code of MAJORITY is simple to write, but it is a bit tricky to analyze. Note that MAJORITY does not sort the entries in A first, but directly finds the majority element (if it exists). The obvious alternative to the approach of MAJORITY above is to sort A first, and then count if $A[\lfloor n/2 \rfloor]$ in the sorted array occurs more than $n/2$ times (why?). A third approach is to find the median first, using one of the methods in Chapter 10 in [CLR], and then count if the median occurs more than $n/2$ times (why?). MAJORITY is better than both of these alternatives.

(a) What is the worst-case run time of MAJORITY, measured by the number of comparisons it makes on elements in A? Justify briefly.

(b) Prove that MAJORITY is correct.
Hint: The heart of the algorithm is the **repeat-until** loop, and a formal proof involves an induction on the number of times this loop is iterated. But here is an observation that may help your intuition. Given a list of integers $x_1, x_2, \ldots, x_n$, if $x_i \neq x_j$ and we remove both x_i and x_j from the list, then a majority in the original list remains a majority in the new list. The converse is not true: The list 5,6,3,3,8,3 has no majority, but if we remove 5 and 6 then the new list 3,3,8,3 has a majority.

(c) What is the purpose of lines 4,5,11,12, and 16, in MAJORITY? Simplify MAJORITY accordingly.

(d) Adjust MAJORITY so that it now finds an element in A which is a 2/3 majority, i.e. an element which occurs more than $2n/3$ times in A, if it exists.
Hint: This is easy.

(e) **Bonus question**: Adjust MAJORITY so that it now finds an element in A which is a 1/3 plurality, i.e. an element which occurs more than $n/3$ times in A, if it exists.
Hint: This is more subtle.

2. Exercise 27.2-10, page 600, in [CLR].
Hint: You may find it helpful to read the proof of Theorem 27.9, page 598. Note in particular the conclusions in the last paragraph on page 598.

3. Problem 27-2, page 626, in [CLR].
Hint for part b: The answer is "no", for the algorithm based on the hint given for part a. You still have to explain why it is "no".

Created: 96.02.29
Modified: 96.03.01