Analysis of Algorithms - Homework I (Solutions)

K. Subramani
LCSEE,
West Virginia University,
Morgantown, WV
{ksmani@csee.wvu.edu}

1 Problems

1. Write a recursive algorithm to compute the maximum element in an array of integers. You may assume the existence of a function "$\max(a,\ b)$" that returns the maximum of *two* integers a and b.

   Solution: □

```
Function FIND-ARRAY-MAX(A, n)
1: if (n = 1) then
2:    return(A[1])
3: else
4:    return(max(A[n], FIND-ARRAY-MAX (A, n − 1) ))
5: end if
```

Algorithm 1.1: Finding the maximum in an array of n elements

2. Argue that your algorithm is correct.

   Solution: We first need to formulate the proposition for algorithm correctness. In this case, we let $P(n)$ stand for the proposition that Algorithm (1.1) finds and returns the maximum integer in the locations $\mathbf{A}[1]$ through $\mathbf{A}[n]$. Accordingly, we have to show that $(\forall n)\ P(n)$ is **true**.

   BASIS: When there is only one element in the array , i.e., $n = 1$, then this element is clearly the maximum element and it is returned on Line 2. We thus see that $P(1)$ is true.

   INDUCTIVE STEP: Assume that Algorithm (1.1) finds and returns the maximum element, when there are exactly k elements in $\mathbf{A}$.

   Now consider the case in which there are $k + 1$ elements in $\mathbf{A}$. Since $(k + 1) > 1$, Line 4 will be executed. In this step, we first make a recursive call to FIND-ARRAY-MAX with exactly k elements. From the inductive hypothesis, we know that the maximum elements in $\mathbf{A}[1]$ through $\mathbf{A}[k]$ is returned. Now the maximum element in $\mathbf{A}$ is either $\mathbf{A}[k + 1]$ or the maximum element in $\mathbf{A}[1]$ through $\mathbf{A}[k]$ (say r). Thus, returning the maximum of $\mathbf{A}[k + 1]$ and r clearly gives the maximum element in $\mathbf{A}$, thereby proving that $P(k) \to P(k + 1)$. By applying the first principle of mathematical induction, we can conclude that $(\forall n)\ P(n)$ is **true**, i.e., Algorithm (1.1) is correct. □

3. What is the *exact* comparison complexity of your algorithm? Derive a recurrence relation and solve it to justify your answer.

   Solution: Observe that the function $\max(a,\ b)$ uses exactly one comparison. Thus, the comparison complexity of Algorithm (1.1) can be described the recurrence relation:

$$\begin{aligned} T(1) &= 0 \\ T(n) &= T(n-1) + 1,\ n > 1 \end{aligned}$$

This recurrence can be expanded as $T(n) = 1 + 1 + \ldots 1 \;\; (n-1)$ times to give $T(n) = n - 1$. □

4. Argue using induction that the exact solution to the recurrence relation:

$$\begin{aligned} T(1) &= 0 \\ T(n) &= 2 \cdot T(\frac{n}{2}) + n, \; n \geq 2 \end{aligned}$$

is $T(n) = n \cdot \log n$.

Solution:

BASIS: At $n = 1$, both the closed form and the recurrence relation agree $(0 = 0)$ and so the basis is true.

INDUCTIVE STEP: Assume that $T(r) = r \cdot \log r$ for all $1 \leq r \leq k$. Now consider $T(k+1)$. As per the recurrence relation, we have,

$$\begin{aligned} T(k+1) &= 2 \cdot T(\frac{k+1}{2}) + (k+1), \text{ since (k + 1)} \geq 2 \\ &= 2 \cdot (\frac{(k+1)}{2} \cdot \log \frac{k+1}{2}) + (k+1) \text{ as per the inductive hypothesis, since } \frac{\text{k}+1}{2} < \text{k} \\ &= (k+1) \cdot [\log(k+1) - \log 2] + (k+1) \\ &= (k+1) \cdot \log(k+1) - (k+1) + (k+1) \\ &= (k+1) \cdot \log(k+1) \end{aligned}$$

We can therefore apply the second principle of mathematical induction to conclude that the exact solution to the given recurrence is $n \cdot \log n$. □

5. Show that $log(n!) \in O(n \cdot \log n)$.

Solution: Observe that

$$\begin{aligned} \log 1 &\leq \log n \\ \log 2 &\leq \log n \\ \vdots &\;\; \vdots \;\; \vdots \\ \log n &\leq \log n \end{aligned}$$

Adding up both the LHS and the RHS, we get

$$\begin{aligned} \Sigma_{i=1}^{n} \log i &\leq n \cdot \log n \\ \Rightarrow \log(1 \cdot 2 \cdot \ldots n) &\leq n \cdot \log n \\ \Rightarrow \log(n!) &\leq n \cdot \log n \end{aligned}$$

Yet another way of proving the above identity is observing that $\log n$ is an increasing function of n and hence we can use the bound

$$\begin{aligned} \log(n!) &= \log 1 + \log 2 + \ldots \log n \\ &= \Sigma_{i=1}^{n} \log i \\ &\leq \int_{1}^{n+1} \log x \; dx \end{aligned}$$

From elementary calculus, we know that $\int \log x = x \log x - x$, from which it follows that $\log(n!) \leq (n+1) \cdot \log(n+1) - n$ and hence $\log(n!) \in O(n \cdot \log n)$. □