Homework 5 sample solution

Due 09/17/15

September 10, 2015

1. Prove that the incorrect sorting algorithm below runs in $O(n \lg n)$ time. (*Hint*: you may use the fact that $\sum_{i=1}^{n} \frac{1}{i} = O(\lg n)$.)

Input: *data*: an array of integers to sort
Input: n: the number of values in data
Output: a permutation of *data* such that
$data[1] \leq data[2] \leq \ldots \leq data[n]$
Algorithm: BadSort
foreach $i = n - 1$ to 1 **step** -1 **do**
  foreach $j = 1$ to $n - i$ **step** i **do**
    if $data[j] > data[j + i]$ **then**
      Swap $data[j]$ and $data[j + i]$
    end
  end
end
return data

Answer:

Proof. Note that the outer **for** loop executes $n - 1$ times. The value of j on the x^{th} iteration of the inner **for** loop is $1 + (x - 1)i$, so the number of iterations will be the largest x that satisfies $1 + (x - 1)i \leq n - i$.

$$\begin{aligned}
1 + (x-1)i &\leq n - i \\
(x-1)i &\leq n - i - 1 \\
x - 1 &\leq \frac{n - i - 1}{i} \\
x - 1 &\leq \frac{n-1}{i} - 1 \\
x &\leq \frac{n-1}{i}
\end{aligned}$$

Consequently, the inner **for** loop executes $\lfloor \frac{n-1}{i} \rfloor = O(\frac{n-1}{i})$ times on iteration i. The instructions inside the inner **for** loop all take $O(1)$ time, for a total time of $O(\frac{n}{i})$. Thus, the two loops execute a total of $\sum_{i=1}^{n-1} O(\frac{n}{i})$ instructions.

$$\begin{aligned}
\sum_{i=1}^{n} \frac{n}{i} &\leq \sum_{i=1}^{n} (\frac{n}{i} + 1) \\
&= n + \sum_{i=1}^{n} \frac{n}{i} \\
&= O(n) + n \sum_{i=1}^{n} \frac{1}{i} \\
&= O(n) + nO(\lg n) \\
&= O(n) + O(n \lg n) \\
&= O(n \lg n)
\end{aligned}$$

The instructions inside the inner **for** loop all take $O(1)$ time, for a total time of $O(n \lg n)O(1) = O(n \lg n)$. The **return** statement takes $O(1)$ time, but this is dominated by the $O(n \lg n)$ time taken by the **for** loops, so BadSort runs in $O(n \lg n)$ time. $\square$