Fundamental Algorithms, Problem Set 3

Due February 8/9, in Recitation

> Well, you see, Haresh Chacha, its like this. First you have ten, that's just ten, that is, ten to the first power. Then you have a hundred, which is ten times ten, which makes it ten to the second power. Then you have a thousand which is ten to the third power. Then you have ten thousand, which is ten to the fourth power - but this is where the problem begins, don't you see? We don't have a special word for that, and we really should. ... But you know, said Haresh, I think there is a special word for ten thousand. The Chinese tanners of Calcutta once told me that they used the number ten-thousand as a standard unit of counting. What they call it I can't remember ... Bhaskar was electrified. But Haresh Chacha you must find that number for me, he said. You must find out what they call it. I have to know, he said, his eyes burning with mystical fire and his small frog-like features taking on an astonishing radiance.
> – from A Suitable Boy by Vikran Seth

1. Write each of the following functions as $\Theta(g(n))$ where $g(n)$ is one of the standard forms: $2n^4 - 11n + 98$; $6n + 43n \lg n$; $63n^2 + 14n \lg^5 n$; $3 + \frac{5}{n}$

2. Illustrate the operation of `RADIX-SORT` on the list: COW, DOG, SEA, RUG, ROW, MOB, BOX, TAB, BAR, EAR, TAR, DIG, BIG, TEA, NOW, FOX following the Figure in the Radix-Sort section. (Use alphabetical order and sort one letter at a time.)

3. Illustrate the operation of `BUCKET-SORT` (with 10 buckets) on the array $A = (.79, .13, .16, .64, .39, .20, .89, .53, .71, .43)$ following the Figure in the Bucket-Sort section.

4. Given $A[1 \cdots N]$ with $0 \le A[I] < N^N$ for all I.
   (a) How long will `COUNTING-SORT` take?
   (b) How long will `RADIX-SORT` take using base N?
   (c) How long will `RADIX-SORT` take using base $N^{\sqrt{N}}$? (Assume $\sqrt{N}$ integral.)

5. Write the time $T(N)$ (don't worry about the output!) for the following algorithms in the form $T(N) = \Theta(g(N))$ for a standard $g(N)$. For time, consider the total number of times X++, I=2*I, J++,J=2*J respectively are applied. (Note: * means multiplication, ++ means increment one.) The hardest is the last one, there is an outer FOR I loop, write the time it takes inside the loop as a function of I and N. Then try (!) to add over $I = 1$ to N.

   (a)
   ```
   X=0
   FOR I=1 TO N
       do FOR J=1 TO N
           X ++
   ```
   (b)
   ```
   I=1
   WHILE I < N
       do I = 2*I
   ```
   (c)
   ```
   FOR I=1 TO N
       do J=1
       WHILE J*J < I
           do J ++
   ```
   (d)
   ```
   FOR I = 1 to N
       J=I
       WHILE J < N
           do J=2*J
   ```

6. Prof. Squander decides to do Bucket Sort on n items with n^2 buckets while his student Ima Hogg decides to do Bucket Sort on n items with $n^{1/2}$ buckets. Assume that the items are indeed uniformly distributed. Assume that Ima's algorithm for sorting inside a bucket takes time $O(m^2)$ when the bucket has m items.

   (a) Argue that Prof. Squander has made a poor choice of the number of buckets by looking analyzing the time of Bucket Sort in his case.

   (b) Argue that Ima has made a poor choice of the number of buckets by looking analyzing the time of Bucket Sort in her case.

   (c) Argue that Ima uses roughly the same amount of *space* as someone using n buckets.

Every universe, our own included, begins in conversation.
– Michael Chabon