FA homework 3

Quentin McGaw (qm301)

09 February 2017

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:**

   (a) $2n^4 - 11n + 98$

   $\Theta(g(n)) = \Theta(n^4)$

   (b) $6n + 43n \lg n$

   $\Theta(g(n)) = \Theta(n)$

   (c) $63n^2 + 14n \lg^5 n$

   $\Theta(g(n)) = \Theta(n^2)$

   (d) $3 + \frac{5}{n}$

   $\Theta(g(n)) = \Theta(1)$

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.)**

   Initial list: COW, DOG, SEA, RUG, ROW, MOB, BOX, TAB, BAR, EAR, TAR, DIG, BIG, TEA, NOW, FOX
   Pass 1: ['SEA', 'TEA', 'MOB', 'TAB', 'DOG', 'RUG', 'DIG', 'BIG', 'BAR', 'EAR', 'TAR', 'COW', 'ROW', 'NOW', 'BOX', 'FOX']
   Pass 2: ['TAB', 'BAR', 'EAR', 'TAR', 'SEA', 'TEA', 'DIG', 'BIG', 'MOB', 'DOG', 'COW', 'ROW', 'NOW', 'BOX', 'FOX', 'RUG']
   Pass 3: ['BAR', 'BIG', 'BOX', 'COW', 'DIG', 'DOG', 'EAR', 'FOX', 'MOB', 'NOW', 'ROW', 'RUG', 'SEA', 'TAB', 'TAR', 'TEA']

   The following Python code was used to generate the several passes:

```python
from operator import itemgetter

def radix_sort_alpha(words):
    l = len(words[0])
    for w in words:
        if len(w) != l:
            raise Exception("Words should have the same length")
    for i in range(l, 0, -1):
        words = sorted(words, key=itemgetter(i - 1))
        words_str = str([''.join(w) for w in words])
        print "PASS "+str(l - i + 1)+": "+words_str
    return words_str

radix_sort_alpha(["COW", "DOG", "SEA", "RUG", "ROW", "MOB", \
                  "BOX", "TAB", "BAR", "EAR", "TAR", "DIG", \
                  "BIG", "TEA", "NOW", "FOX"])
```

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.**

   Initial input array A:
   [0.79, 0.13, 0.16, 0.64, 0.39, 0.2, 0.89, 0.53, 0.71, 0.43]
   Initial output buckets array B:
   [[], [], [], [], [], [], [], [], [], []]
   Output buckets array B with elements in buckets:
   [[], [0.13, 0.16], [0.2], [0.39], [0.43], [0.53], [0.64], [0.79, 0.71], [0.89], []]
   Output buckets array B with elements sorted in buckets:
   [[], [0.13, 0.16], [0.2], [0.39], [0.43], [0.53], [0.64], [0.71, 0.79], [0.89], []]

Final output array B:
[0.13, 0.16, 0.2, 0.39, 0.43, 0.53, 0.64, 0.71, 0.79, 0.89]
The following Python code was used to generate these stages:

```
from math import floor

def bucket_sort(A):
    print "Initial input array A: "+str(A)
    n = len(A)
    for i in range(n):
        assert(A[i] >= 0 and A[i] < 1)
    B = [[] for _ in range(n)]
    print "Initial output buckets array B: "+str(B)
    for i in range(n):
        place = int(floor(A[i] * n))
        B[place].append(A[i])
    print "Output buckets array B with \
               elements in buckets: "+str(B)
    for j in range(n):
        B[j].sort()
    print "Output buckets array B with \
               elements sorted in buckets: "+str(B)
    B_final = []
    for bucket in B:
        B_final += bucket
    print "Final output array B: "+str(B_final)
    return B_final

bucket_sort([.79,.13,.16,.64,.39,.20,.89,.53,.71,.43])
```

4. **Given $A[1 \cdots N]$ with $0 \le A[I] < N^N$ for all I.**

   (a) **How long will `COUNTING-SORT` take?**
   For `COUNTING-SORT`, $n = N$ and $k = N^N$. Hence it will take $O(n + k) = O(N^N)$ time

   (b) **How long will `RADIX-SORT` take using base N?**
   For `RADIX-SORT`, $n = N$, $k = N^N$ and $b = N$. Hence it will take $O((n+b)log_b k) = O((N+N)log_N N^N) = O((N+N)N) = O(N^2)$ time

   (c) **How long will `RADIX-SORT` take using base $N^{\sqrt{N}}$? (Assume $\sqrt{N}$ integral.)**
   For `RADIX-SORT`, $n = N$, $k = N^N$ and $b = N^{\sqrt{N}}$. Hence it will take $O((n+b)log_b k) = O((N+N^{\sqrt{N}})log_{N^{\sqrt{N}}} N^N) = O((N+N^{\sqrt{N}})2 = O(N^{\sqrt{N}})$ time

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 ++

   $T(N) = \Theta(N^2)$

   (b) **I=1**
   WHILE I < N
   do I = 2*I

   $T(N) = \Theta(log_2 N)$

   (c) **FOR I=1 TO N**
   do J=1
   WHILE J*J < I
   do J ++

   $T(N) = \Theta(NlogN)$

   (d) **FOR I = 1 to N**
   J=I
   WHILE J < N
   do J=2*J

   $T(N) = \Theta(N)$

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.**

   For Prof. Squander, we have $n = n$ and $k = n^2$ so we require $\Theta(n + n^2) = \Theta(n^2)$ time, which is the worst case scenario for a bucket sort (upper bond).

(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.**

For Ima Hogg's bucket sort, because of the uniform distribution, there should be $n^{\frac{1}{2}}$ items per bucket so sorting each bucket takes $\Theta((n^{\frac{1}{2}})^2) = \Theta(n)$ thus we require $\Theta(n^{\frac{3}{2}})$ which is still worst than $\Theta(n + k)$.

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

Because there is $n^{\frac{1}{2}}$ items per bucket and there are $n^(1/2)$ buckets, the total space used is $n^{\frac{1}{2}} * n^{\frac{1}{2}} = n$ which is obviously the same as the space used by someone using n buckets.

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