Vivek Wadhwani

vivek121

EECS 2011 – Assignment #1

1) Perform an analysis of each of the following fragments and give a θ bound for the running time:

```
1a. $\boldsymbol{sum = 0;}$                                      $---1\ unit\ time$
        $\boldsymbol{for\ (int\ i = 0;\ i < N;\ i++)}$          $---untangle$
            $\boldsymbol{for\ (int\ j = i; j \geq 0; j--)}$      $---untangle$
                $\boldsymbol{sum\ ++;}$                        $---1\ unit\ time$
```

We can untangle the loops following the table below:

i	j
0	0
1	1,0
2	2,1,0
3	3,2,1,0
4	4,3,2,1,0
.....	......
$N-1$	$N-1, N-2, \ldots\ldots, 3, 2, 1, 0$
	Sum of first N numbers: $\frac{(N)(N+1)}{2} = \frac{N^2}{2} + \frac{N}{2}$

Using the information above,

$$\therefore T(N) = 1_{sum=0} + \left(\frac{N^2}{2} + \frac{N}{2}\right)_{for\ loops} \times 1_{sum++}$$

$$\rightarrow T(N) = \frac{N^2}{2} + \frac{N}{2} + 1$$

Since running time is a **Polynomial** running time hence its $\boxed{\text{theta bound is } \theta(N^2)}$

```
1b. $\boldsymbol{sum = 0;}$                                  $--- 1\ unit\ time$
        $\boldsymbol{for\ (int\ i = 0;\ i < N;\ i++)}$           $--- untangle$
                $\boldsymbol{for\ (int\ j = 0; j < i * i; j++)}$     $--- untangle$
                        $\boldsymbol{sum++;}$                  $--- 1\ unit\ time$
```

We can untangle the loops using the table below:

i	j
0	X
1	1,0
2	3,2,1,0
3	8,7,6,5,4,3,2,1,0
.....	......
$N-1$	$(N-1)^2 - 1, \dots \dots, 2, 1, 0$
	Sum of first $(N-1)^2$ numbers: $\frac{(N-1)(N)(2N-1)}{6} = \frac{N^3}{3} - \frac{N^2}{2} + \frac{N}{6}$

Using the information above:

$$\therefore T(N) = 1 + \left(\frac{N^3}{3} - \frac{N^2}{2} + \frac{N}{6}\right) \times 1$$

$$\rightarrow T(N) = \frac{N^3}{3} - \frac{N^2}{2} + \frac{N}{6} + 1$$

Since running time is a **Polynomial** running time with degree 3 hence its $\boxed{\text{theta bound is } \theta(N^3)}$

2) Show that:

2a. $2^{N+1} is\ O(2^N)$

We know that, $2^{N+1} = 2^N \times 2 \quad and \quad f(n)\ is\ O\big(g(n)\big) \quad if \quad f(n) \le c \times g(n)$

$\therefore \ 2^{N+1} \le 2^N \times 2$

$\rightarrow \ 2^{N+1} \le 2^N \times c \qquad [for\ c = 2\ and\ N \ge N_0]$

$\boxed{\text{Thus, } 2^{N+1}\ is\ O(2^N)}$

2b. $N^2/_2 + N + 10\ \ is\ \Omega(N^2)$

We know that, $f(n)\ is\ \Omega\big(g(n)\big) \quad if \quad f(n) \ge c \times g(n)$

$\frac{N^2}{2} + N + 10 \ \ge\ \frac{1}{2} \times N^2 \qquad \left[For\ c = \frac{1}{2}\ and\ N \ge 0\right]$

$\rightarrow \frac{N^2}{2} + N + 10 \ \ge\ \ c \times N^2$

$\boxed{\therefore \ N^2/_2 + N + 10\ \ is\ \Omega(N^2)}$

2c. $N^{1.5}$ *grows faster than NlogN using L'Hopital's rule.*

Let $f(N) = N^{1.5} \quad and \quad g(N) = NlogN$

L'Hopital rule states that : $if \ \lim_{n\to\infty} f(N) = \infty \quad and \quad \lim_{n\to\infty} g(N) = \infty$

$$then, \quad \lim_{n\to\infty}\frac{f(N)}{g(N)} = \lim_{n\to\infty}\frac{f'(N)}{g'(N)}$$

Using L'Hopital rule on the above functions:

$$\lim_{n\to\infty}\frac{f(N)}{g(N)} = \lim_{n\to\infty}\frac{N^{1.5}}{NlogN} = \lim_{n\to\infty}\frac{\cancel{N}\sqrt{N}}{\cancel{N}logN} = \lim_{n\to\infty}\frac{\sqrt{N}}{logN} \quad \text{[deriving the numerator and denominator]}$$

$$\rightarrow \ \lim_{n\to\infty}\frac{1/_{2\sqrt{N}}}{1/_N} = \boxed{\lim_{n\to\infty}\frac{\sqrt{N}}{2} = \infty}$$

The infinity implies that $g(N)\ \ is\ \ o\big(f(N)\big)$

$\rightarrow NlogN \quad is \quad o(N^{1.5})$

$\boxed{\therefore We\ can\ say\ that\ N^{1.5}\ grows\ faster\ than\ NlogN}$

2d. $log^k N$ is o(N) for any constant k. (Hint: use L'Hopital Rule)

Let $f(N) = N \quad and \quad g(N) = log^k N$

L'Hopital rule states that : $if \ \lim_{n\to\infty} f(N) = \infty \quad and \quad \lim_{n\to\infty} g(N) = \infty$

$$then, \ \lim_{n\to\infty} \frac{f(N)}{g(N)} = \lim_{n\to\infty} \frac{f'(N)}{g'(N)}$$

Using L'Hopital rule on the above functions:

$$\lim_{n\to\infty} \frac{f(N)}{g(N)} = \lim_{n\to\infty} \frac{log^k N}{N} \qquad [deriving\ the\ numerator\ and\ the\ denominator]$$

$$\lim_{n\to\infty} \frac{\frac{klog^{k-1}N}{N}}{1} = \lim_{n\to\infty} \frac{klog^{k-1}N}{N} \qquad [deriving\ again]$$

$$\lim_{n\to\infty} \frac{\frac{k(k-1)log^{k-2}N}{N}}{1} = \lim_{n\to\infty} \frac{k(k-1)log^{k-2}N}{N} \qquad [deriving\ again]$$

Let us keep decreasing the value of k up to $k-1$

$$\lim_{n\to\infty} \frac{\frac{k(k-1).....(k-(k-1))log^{k-(k-1)}N}{N}}{1} = \lim_{n\to\infty} \frac{k(k-1).....2.1(log^1 N)}{N}$$

$$\to \lim_{n\to\infty} \frac{k(k-1).....2.1(log^1 N)}{N} = (k(k-1)\ .....2.1) \lim_{n\to\infty} \frac{(logN)}{N} \quad [deriving\ again]$$

$$\to \lim_{n\to\infty} \frac{1}{N} = 0 \ \boxed{\to f(N)\ is\ o(g(N))}$$

$\boxed{\therefore We\ can\ say\ that\ log^k N\ is\ o(N)}$

2e. 2^{10N} *is not* $O(2^N)$

Let, $f(N) = 2^{10N} \qquad and \qquad g(N) = 2^N$

We know that, $\quad f(n)\ is\ O(g(n)) \quad if \quad f(n) \le c \times g(n)$

In the equation above we substitute the value of $f(N)\ and\ g(N) \qquad [for\ N \ge N_0]$

$$\to 2^{9N} \times 2^N \le c \times 2^N$$

$$\to 2^{9N} \le c \qquad [taking\ \log_2\ on\ both\ the\ sides]$$

$$\to 9N \le log_2(c)$$

$$\boxed{\to N \le {}^{log_2(c)}/_9 \qquad [This\ implies\ that\ N\ is\ less\ than\ a\ constant\ value]}$$

Since N changes its value constantly, no values of $c\ and\ N_0$ can satisfy the condition

$$f(n) \le c \times g(n) \qquad [for\ N \ge N_0]$$

$$\therefore\ 2^{10N}\ \ is\ not\ O(2^N)$$

3) Solve the following recurrences by obtaining a θ bound for T(N) given that T(1) = θ(1)

3a. $\boldsymbol{T(N) = 2N - 1 + T(N-1)}$

We start by looking for values of T(N) as we keep decreasing the value of n:

$T(N-1) = 2(N-1) - 1 + T(N-2) = 2N - 3 + T(N-2)$

$T(N-2) = 2(N-2) - 1 + T(N-3) = 2N - 5 + T(N-3)$

$T(N-3) = 2(N-3) - 1 + T(N-4) = 2N - 7 + T(N-4)$

Now we use the above values and substitute it back in the main equation

$T(N) = 1(2N-1) + T(N-1) \qquad --- (1) \quad [We\ get\ \ T(N-1)\ from\ above]$

$T(N) = 2N - 1 + 2N - 3 + T(N-2)$

$T(N) = 2(2N-2) + T(N-2) \qquad --- (2)$

$T(N) = 4N - 4 + 2N - 5 + T(N-3)$

$T(N) = 3(2N-3) - 1 + T(N-3) \qquad --- (3)$

$T(N) = 6N - 7 + 2N - 7 + T(N-4)$

$T(N) = 4(2N-4) + T(N-4) \qquad --- (4)$

From (1), (2),(3) and (4) Here we can establish a pattern that is:

$\boldsymbol{T(N) = k(2N-k) + T(N-k)}$

Now we assume that $\mathbf{N = k + 1}$,

$T(k) = k(2k + 2 - k) + T(k + 1 - k)$

$T(k) = k^2 + 2k + T(1) \qquad [Since\ k = N - 1]$

$\boxed{T(N) = (N-1)^2 + 2(N-1) + \theta(1)}$

Since T(N) has a polynomial running time of degree 2 its $\boxed{\text{theta bound is } \theta(N^2)}$

3b. $\boldsymbol{T(N) = N + T(N-3)}$

We start by looking for values of T(N) as we keep decreasing the value of n:

$T(N-3) = N - 3 + T(N-6)$

$T(N-6) = N - 6 + T(N-9)$

$T(N-9) = N - 9 + T(N-12)$

Using the above values we substitute it back in the main equation:

$T(N) = (1 \times N) - (3 \times (0)) + T(N-3) \quad --- (1)$

$T(N) = (2 \times N) - (3 \times (1+0)) + T(N-6) \quad --- (2)$

$T(N) = (3 \times N) - (3 \times (0+1+2)) + T(N-9) \quad --- (3)$

$T(N) = (4 \times N) - (3 \times (0+1+2+3)) + T(N-12) \quad --- (4)$

From (1),(2),(3) and (4) we can establish a pattern here:

$$\boldsymbol{T(N) = (k \times N) - \left(3 \times \frac{k \times (k-1)}{2}\right) + T(N-3k)}$$

To get T(1) we assume, $N - 3k = 1 \rightarrow N = 1 + 3k$:

$\mathrm{T}(1+3\mathrm{k}) = (\mathrm{k} \times (3\mathrm{k}+1)) - \left(3 \times \frac{\mathrm{k} \times (\mathrm{k}-1)}{2}\right) + \mathrm{T}(1) \quad [\mathrm{T}(1) = \theta(1)]$

$\mathrm{T}(1+3\mathrm{k}) = \frac{3k^2}{2} + \frac{5k}{2} + \theta(1) \qquad [Since\ k = \frac{N-1}{3}]$

$\mathrm{T(N)} = \frac{3(\frac{N-1}{3})^2}{2} + \frac{5(\frac{N-1}{3})}{2} + \theta(1)$

$$\boxed{\mathrm{T(N)} = {(N^2 + 3N - 4)}/{6} + \theta(1)}$$

Since T(N) has a polynomial running time of degree 2 its $\boxed{\text{theta bound is } \theta(N^2)}$

3c. $\mathbf{T(N) = N^2 + T(N-1)}$

We start by looking for values of T(N) as we keep decreasing the value of n:

$\mathrm{T}(\mathrm{N}-1) = (N-1)^2 + T(N-2)$

$\mathrm{T}(\mathrm{N}-2) = (N-2)^2 + T(N-3)$

$\mathrm{T}(\mathrm{N}-3) = (N-3)^2 + T(N-4)$

Using the above values we substitute it back in the main equation:

$\mathrm{T}(\mathrm{N}) = T(N-1) + N^2 \qquad ---(1)$

$\mathrm{T}(\mathrm{N}) = T(N-2) + N^2 + (N-1)^2 \qquad ---(2)$

$\mathrm{T}(\mathrm{N}) = T(N-3) + N^2 + (N-1)^2 + (N-2)^2 \qquad ---(3)$

$\mathrm{T}(\mathrm{N}) = T(N-4) + N^2 + (N-1)^2 + (N-2)^2 + (N-3)^2 \qquad ---(4)$

From (1),(2),(3) and (4) we can establish a pattern here:

$\boldsymbol{T(N) = T(N-k) + N^2 + (N-1)^2 + \ldots\ldots + (N-k+2)^2 + (N-k+1)^2}$

To get the T(1) in the equation we assume that $\boldsymbol{N = k+1}$ *:*

$T(N) = T(1) + N^2 + \cdots .. + (3)^2 + (2)^2 + (1)^2 \qquad [\mathrm{T}(1) = \theta(1)]$

$\boxed{\mathrm{T}(\mathrm{N}) = \theta(1) + {}^{N(N+1)(2N+1)}/_{6} \qquad [Sum\ of\ first\ N^2\ terms]}$

Since T(N) has a polynomial running time of degree 3 its $\boxed{theta\ bound\ is\ \theta(N^3)}$

4) Mergesort does have a worst-case time of $\theta(NlogN)$ but its overhead (hidden in the constant factors) is high and this is manifested near the bottom of the recursion tree where many merges are made. Someone proposed that we stop the recursion once the size reaches K and switch to insertion sort at that point. Analyze this proposal (by modifying the recurrence analysis of standard mergesort) and prove that its running time is $\theta(NK + Nlog(\frac{N}{K}))$.

We start by defining the running time for the recurrence relation of Merge Sort:

and We know that from the slides,

$$T(N) = 2^c T\left(\frac{N}{2^c}\right) + cN \qquad [where\ c\ is\ a\ constant] \qquad ---(1)$$

As mentioned above we should tune the mergesort algorithm to switch to insertion sort when the array size reaches less than equal to ($\leq$) k. For any array value size below k we perform insertion sort which takes running time of k^2

$$\rightarrow\ T(k) = k^2$$

To find the theta (θ) growth rate can be found when the value of k is,

$$k = \frac{N}{2^c} \rightarrow 2^c = \frac{N}{k} \qquad \text{[performing logarithim on both sides]}$$

$$c = log_2\left(\frac{N}{k}\right)$$

Substituting the value of c above in equation (1) we get,

$$T(N) = 2^{log_2\left(\frac{N}{k}\right)} T\left(\frac{N}{2^{log_2\left(\frac{N}{k}\right)}}\right) + cN$$

$$\rightarrow\ T(N) = (\tfrac{N}{k})T\left(\frac{N}{\frac{N}{k}}\right) + Nlog_2\left(\frac{N}{k}\right)$$

$$\rightarrow T(N) = (\tfrac{N}{k})T(k) + Nlog_2\left(\frac{N}{k}\right) \qquad [we\ know\ that\ T(k) = k^2]$$

$$\rightarrow T(N) = (\tfrac{N}{k})k^2 + Nlog_2\left(\frac{N}{k}\right)$$

$$\rightarrow T(N) = Nk + Nlog_2\left(\frac{N}{k}\right)$$

$\therefore$ The theta (θ) growth rate can be found by saying $\theta\big(T(N)\big)$,

$$\boxed{\rightarrow\ \theta\ (Nk + Nlog_2\left(\tfrac{N}{k}\right))}$$