EECS 2011 - Assignment 1

Gurban Jumaqulyyev
Student #: 216151268

Question 1

1a)

```
sum = 0;                                    ln 1 - 1 unit time
for (int i=0; i<N; i++)                     ln 2 - ??
    for (int j=i; j≥0; j--)                 ln 3 - ??
        sum++;                              ln 4 - 1 unit time
```

In order to get the running time of the nested loop, we must untangle the loop using the method 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, ..., 3, 2, 1, 0

from this we could get the sum of first N numbers; which is the following:

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

Therefore, from the untangling above we could say that:

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

so, whe we simplify we get: $T(N) = \frac{N^2}{2} + \frac{N}{2} + 1$

As we know that running time is a Polynomial, we found that the θ bound is:

$$\boxed{\theta(N^2)}$$

Gurban Jumagulyyev
21615268

1b)

```
sum = 0;                                    ln 1  -  1 unit time
for (int i=0; i<N; i++)                     ln 2  -  ??
    for (int j=0; j<i*i; j++)               ln 3  -  ??
        sum++;                              ln 4  -  1 unit time
```

To get the running time of the nested loop, we must untangle the loop:

i	j
0	X
1	1, 0
2	3, 2, 1, 0
3	8, 7, 6, 5, 4, 3, 2, 1, 0
....	....
N - 1	$(N-1)^2-1, \ldots, 2, 1, 0$

from this we can get the sum of the first $(N-1)^2$ numbers, which will be:

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

Therefore, from untangling above, we can say the following:

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

so, when we simplify: $T(N) = \frac{N^3}{3} - \frac{N^2}{2} + \frac{N}{6} + 1$

As we can see, the running time is a Polynomial which has the highest degree of 3. So, we can say that the θ bound is:

$$\boxed{\theta(N^3)}$$ //

Gurban Jumagulyyev
21615126 8

Question 2

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

we know that $2^{N+1} = 2^N \times 2$
we also know that $f(n)$ is $O(g(n))$ if $f(n) \leq c \times g(n)$

therefore we can say: $2^{N+1} \leq 2^N * 2$

so, for $c = 2$ and $n \geq n_0 \Rightarrow 2^{N+1} \leq 2^N * c$

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

2b) $\frac{N^2}{2} + N + 10$ is $\Omega(N^2)$

if $f(n) \geq c * g(n)$, we know that $f(n)$ is $\Omega(g(n))$

so, for $c = \frac{1}{2}$ and $N \geq 0 \Rightarrow \frac{N^2}{2} + N + 10 \geq \frac{1}{2} \times N^2$

Hence, the general eq: $\frac{N^2}{2} + N + 10 \geq c \times N^2$

Therefore, we showed that $\boxed{\frac{N^2}{2} + N + 10 \text{ is } \Omega(N^2)}$

2c) $N^{1.5}$ grows faster than $N \log N$ using L'Hopital's rule.

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

then $\lim_{n \to \infty} \frac{f(N)}{g(N)} = \lim_{n \to \infty} \frac{f'(N)}{g'(N)}$ cont. on next page

Gurban Jumagulyyev
21615l268

so, now we let $f(N) = N^{1.5}$ and $g(N) = N\log N$

then we can apply L'Hopitals rule to these fonctions:

$$\lim_{n\to\infty} \frac{f(N)}{g(N)} = \lim_{n\to\infty} \frac{N^{1.5}}{N\log N} = \lim_{n\to\infty} \frac{\cancel{N}\sqrt{N}}{\cancel{N}\log N} = \lim_{n\to\infty} \frac{\sqrt{N}}{\log N}$$

after applying L'Hopital's rule: $\lim_{n\to\infty} \frac{\frac{1}{2}\sqrt{N}}{1/N} = \boxed{\lim_{n\to\infty} \frac{\sqrt{N}}{2} = \infty}$

what the answer implies is that $g(N)$ is $o(f(N))$

$N\log N$ is $o(N^{1.5})$

From above we can conclude that $N^{1.5}$ grows faster than $N\log N$

2d) $\log^k N$ is $o(N)$ for any constant k.

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

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

so, we let $f(n) = \log^k N$ and $g(n) = N$

Applying L'Hopital's rule: $\lim_{n\to\infty} \frac{k\log^{(k-1)} N \times \frac{1}{N}}{1}$

differentiating: $\lim_{n\to\infty} \frac{k(k-1)\log^{(k-2)} N \times \frac{1}{N}}{1}$

so, we differentiate until the power of log get to 0:

$\lim_{n\to\infty} \frac{K(k-1)(k-2)\cdots(k-(k+1))\log^{(k-K)} N}{N}$

cont. on next page (5)

Gurban Jumagulyyev
216151268

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

we know that $f(n)$ is $O(g(n))$ only if $f(n) \leq c \times g(n)$

So, we can let $f(N) = 2^{10N}$ and $g(N) = 2^N$

Now for $N > N_0$ we can substitute the values that we set $f(N)$ and $g(N)$ to be into $f(n) \leq c \times g(n)$:

$$\Rightarrow 2^{9N} \times \cancel{2^N} \leq c \times \cancel{2^N}$$

$$\Rightarrow 2^{9N} \leq c$$

from here we can take the $\log_2$ on both the sides

$$\Rightarrow 9N \leq \log_2(c)$$

$$\Rightarrow \boxed{N \leq \frac{\log_2(c)}{9}}$$

so we can see than N is less than a constant value As N is changing its value constantly it cannot be bounded to a constant. This proves that $\boxed{2^{10N} \text{ is not } O(2^N)}$

2d continuation) $\lim_{n \to \infty} \frac{k(k-1)(k-2)\ldots 1}{c} \times \frac{\log^{\circ} N}{N}$

$$\lim_{n \to \infty} \frac{1}{N} \Rightarrow \boxed{0}$$

From the calculations that were done, we can say that $f(n) < cg(n)$ so we can conclude that:

$\boxed{f(n) \text{ is } o(g(n))}$

Gurban Jumagulyyev
2161512 68

Question 3

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

We can start by decreasing the value of n and get the resultant values of T(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 substitute the resultant values we got into the main equation:

$T(N) = 1(2N-1) + T(N-1)$ eq 1
$T(N) = 2N-1 + 2N-3 + T(N-2)$
$T(N) = 2(2N-2) + T(N-2)$ } eq 2
$T(N) = 4N-4 + 2N-5 + T(N-3)$
$T(N) = 3(2N-3) - 1 + T(N-3)$ } eq 3
$T(N) = 6N-7+2N-7+T(N-4)$
$T(N) = 4(2N-4) + T(N-4)$ } eq 4

from the 4 simplified equations above we observe a pattern, which is the following: $T(N) = k(2N-k) + T(N-k)$

Now we can say that $N = k+1$, so we get:
$T(k) = k(2k+2-k) + T(k+1-k)$
$T(k) = k^2 + 2k + T(1)$ and now since $k = N-1$

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

Since the greatest degree of the polynomial is 2, the θ bound is: $\boxed{\theta(N^2)}$

Gurban Jumagulyyev
21615l268

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

we can start by decreasing the value of n and get the resultant values of T(N):

$T(N-3) = N-3 + T(N-6)$
$T(N-6) = N-6 + T(N-9)$
$T(N-9) = N-9 + T(N-12)$

Now substitute the values into the main equation:

eq 1: $T(N) = (1 \times N) - (3 \times 0) + T(N-3)$
eq 2: $T(N) = (2 \times N) - (3 \times 1) + T(N-6)$
eq 3: $T(N) = (3 \times N) - (3 \times 3) + T(N-9)$
eq 4: $T(N) = (4 \times N) - (3 \times 6) + T(N-12)$

from the equations above, we observe the following:

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

we can say that $N-3k=1$ so $N = 1+3k$ in order for us to get T(1):

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

$T(1+3k) = \frac{3k^2}{2} + \frac{5k}{2} + \theta(1)$ and now since $k = \frac{N-1}{3}$

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

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

Since the greatest degree of the polynomial is 2, then the theta (θ) bound is: $\boxed{\theta(N^2)}$

Gurban Jumagulyyev
21615126 8

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

We can start by decreasing the value of n and get the resultant values of T(N):

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

Now we substitute the values into the main equation:

eq 1: $T(N-1) + N^2$
eq 2: $T(N-2) + N^2 + (N-1)^2$
eq 3: $T(N-3) + N^2 + (N-1)^2 + (N-2)^2$
eq 4: $T(N-4) + N^2 + (N-1)^2 + (N-2)^2 + (N-3)^2$

from the four equations above, we observe the following:
$T(N) = T(N-k) + N^2 + (N-1)^2 + \dots + (N-k+2)^2 + (N-k+1)^2$

Now we can say that $N = k+1$ in order for us to get T(1):

$$T(N) = T(1) + N^2 + \dots + (3)^2 + (2)^2 + (1)^2$$

so the sum of the first N^2 terms is as follows:

$$\boxed{T(N) = \theta(1) + \frac{N(N+1)(2N+1)}{6}}$$

Since the greatest degree of the polynomial is 3, then the θ bound is:

$$\boxed{\theta(N^3)}$$

Gurban Jumagulyyev
216151268

Question 4

Mergesort has the following T(N):

$$T(N) = 2T\left(\frac{N}{2}\right) + N$$

$$T(N) = 2\left(2T\left(\frac{N}{4}\right) + \frac{N}{2}\right) + N = 4T\left(\frac{N}{4}\right) + 2N$$

$$T(N) = 4\left(2T\left(\frac{N}{8}\right) + \frac{N}{4}\right) + 2N = 8T\left(\frac{N}{8}\right) + 3N$$

So, now we can observe a gereral equation from from above for i times:

$$T(N) = 2^i T\left(\frac{N}{2^i}\right) + iN$$

we can asume that $k = \frac{N}{2^i}$ and from that we also know that $2^i = \frac{N}{k}$, now we can solve for i:

$$\log 2^i = \log\left(\frac{N}{k}\right)$$

$$\Rightarrow i = \log\left(\frac{N}{k}\right)$$

$$\Rightarrow T(N) = 2^{\log\left(\frac{N}{k}\right)} * T(k) + \log\left(\frac{N}{k}\right) * N$$

T(k) has a running time of insertionsort, which is the following: $O(k^2)$

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

$$\boxed{T(N) = Nk + N\log\left(\frac{N}{k}\right)}$$

We have just proved that T(N) has running time: $\boxed{\theta\left(NK + N\log\left(\frac{N}{k}\right)\right)}$