CS5800: Algorithms — Iraklis Tsekourakis

Homework 1 Name: Avinash R. Arutla

1. (18 points)

1. $n^2 + 7n + 1$ is $\Omega(n^2)$

   Solution:

   To prove that $n^2 + 7n + 1$ is $\Omega(n^2)$,

   We first need to prove the condition that

   f(n) $\geq c.g(n)$

   where f(n) = $n^2 + 7n + 1$, g(n) = n^2 (Since n^2 is the highest power in f(n))

   Assuming,
   c = 1, we can write

   $$n^2 + 5n + 1 \geq n^2$$

   $$7n + 1 \geq 0$$

   $$7n \geq -1$$

   $$n \geq -1/7$$

   For $n_0 = 1$,

   $$n^2 + 7n + 1 \geq n^2$$

   $$1 + 7 + 1 \geq 1^2$$

   $$9 \geq 1$$

   Here, for n_0, for all values greater than 0, the f(n) will stay positive.
   Therefore at c=1, n_0=1, the above equation holds true.

2. $3n^2 + n - 10$ **is** $O(n^2)$

   Solution:

   To prove that $3n^2 + n - 10$ is $O(n^2)$

We need to satisfy that f(n) ≤ c.g(gn) for all $n \geq n_0$

where f(n) = $3n^2 + n - 10$, g(n) = n^2

Here we need to prove that $3n^2 + n - 10 \leq c.n^2$

Since the function $c.g(n)$ must be greater than f(n), we need to cvhoose something which is greater than $3n^2$.

So lets assume, c=4

$$3n^2 + n - 10 \leq 4n^2$$
$$n - 10 \leq n^2$$

Here we can assume that for any value of $n_0 \geq 1$, the positivbe condition holds.

Proof

$$n = 1- > 1 - 10 \leq 1^2 = -9 \leq 1$$
$$n = 2- > 2 - 10 \leq 4 = -8 \leq 4$$
$$n = 1- > 3 - 10 \leq 9 = -7 \leq 9$$
$$n = 1- > 4 - 10 \leq 16 = -6 \leq 16$$

Therefore, for all conditions, we could say

$$3n^2 + n - 10 is O(n^2)$$

3. n^2 **is** $\Omega(nlogn)$
   Solution:

   To prove that n^2 is $\Omega(nlogn)$, we need to prove that $f(n) \geq c.g(n)$ for all $n \geq n_0$

   Here, $f(n) = n^2$, $g(n) = n.logn$

   So, to prove that $n^2 \geq c.(nlogn)$
   Lets assume c=1

   $n^2 \geq 1.(nlogn)$
   $n^2 \geq nlogn$

   From the above equations, we can clearly observe that n^2 grows faster than $nlogn$.

Lets assume $n_0 = 1$

$$1 \geq 1.log1$$

$$1 \geq 1$$

Lets assume $n_0 = 2$

$$2^2 \geq 2.log2$$

$$4 \geq 2$$

Lets assume $n_0 = 1$

$$3^2 \geq 3.log3$$

$$9 \geq 3$$

Therefore we can clearly observe that, at $c = 1$ and $n_0 = 2$, the above expression holds true

2. (20 points)

1. $T(n) = 2T(n/2) + b$ if $n \geq 1$,else 1 if $n = 1$
   Solution:

   The recurrence relation is as follows

$$T(n) = 2T(\frac{n}{2} + b)$$

2nd iteration

$$T(n) = 2[2T\frac{n}{2^2} + b] + b = 2^2T[\frac{n}{2^2}] + 2b + b$$

3rd iteration

$$T(n) = 2^2[2T\frac{n}{2^3} + b] + 3b = 2^3T[\frac{n}{2^3}] + 7b$$

4th iteration

$$T(n) = 2^3[2T\frac{n}{2^4} + b] + 7b = 2^4T[\frac{n}{2^4}] + 15b$$

From the above equation, we can understand the pattern that is observed,

$$T(n) = 2^kT\frac{n}{2^k} + (2^k - 1)b$$

Since the condition is,
T(1) = 1, we stop when
$\frac{n}{2^k} = 1$
$n = 2^k$
$k = log_2 n$

Plugging this back into the equation we get,

$$T(n) = 2^(log_2 n) + \frac{n}{2^(log_2 n)} + (2^(log_2^n - 1)) + b$$

$$= 2^(log_2 n)(1) + (n-1)b$$

$$T(n) = n(1) + (n-1)b$$

Asympotitc notations are as follows

Big-O : The dominant term here is bn, bt sicne we ignore the constants it is O(n)
Big-Ω : Sinmce it atleast grows linearly with n, the lower bound notattion is $\Omega(n)$

2. $T(n) = T(n-1) + n + b$ if $n \geq 1$,else c if $n = 0$
   Solution:

   Solving for the above equation, T(c) = 0

$$T(n-1) = T(n-2) + (n-1) + b$$

1st Iteration

$$T(n-1) + n + b if n > 1$$

2nd Iteration

$$T(n-2) + T(n-1) + n + b + b$$

3rd Iteration

$$T(n-3) + T(n-2) + T(n-1) + n + 3b$$

4th Iteration

$$T(n-4) + T(n-3) + T(n-2) + T(n-1) + n + 4b$$

The pattern here is,

$$T(n) = T(n-k) + \sum_{i=1}^{k}(n-k-i) + kb$$

For the base case, where T(n-k) =0,
n-k=0
n=k at n=0

$$T(n) = T(n-n) + \sum_{i=1}^{n}(n-n-i) + nb$$

$$T(n) = c + \sum_{i=1}^{n}(-i) + nb$$

$$T(n) = c + \frac{n(n+1)}{2} + nb$$

$$T(n) = c + \frac{n^2}{2} + \frac{n}{2} + nb$$

$$T(n) = \frac{n^2}{2} + \frac{n}{2} + nb + c$$

From here, we can understanmd that order of growth is $O(n^2)$

3. (20 points)

1. $T(n) = T(n-3) + 3logn$

   Solution:

   Our guess here is that, T(n) = (nlogn)

   Here we need to show that $T(n) \leq cnlog$

   $$T(n) \leq c(n-3)log(n-3) + 3logn$$

   $$T(n) \leq cnlog(n-3) - 3clog(n-3) + 3logn$$

   Here, we consider -3 to be negligible compared to log(n)

   So, $log(n) > log(n-3)$

   So we can write here that,

   $$T(n) \leq cnlog(n) - 3clogn + 3logn$$

   Therefore, $T(n) \leq cnlogn(n)$ [Neglecting all the coeffecients]

   Here, we can write,

   $$-3clogn + 3logn \leq 0$$

   $$3logn(1-c) \leq 0$$

   $$c \geq 1$$

   Therefore, we proved that for a constant $c \geq 1$, $T(n) \leq cnlog(n)$, and therefore $T(n) = O(logn)$

2. $T(n) = 4T(\frac{n}{3}) + n$

   Solution:

   Here our guess is $T(n) = O(nlog_3 4)$

   We need to show that, $T(n) \leq cn^{(\log_3 4)}$

Given that, $T(n) = 4T(n/3) + n$

We can write,

$$T(n) \leq cn^{(\log_3 4)}$$
$$4.T(n) \leq c\frac{n}{3}^{(\log_3 4)} + n$$
$$4.T(n) \leq c\frac{n^{(\log_3 4)}}{3^{(\log_3 4)}} + n$$
$$4.T(n) \leq c\frac{n^{(\log_3 4)}}{4} + n$$
$$T(n) \leq cn^{(\log_3 4)} + n$$

Here, we need to prove that $T(n) \leq c[n^{log_3 4}] + n$, this will hold for an appropriate choice of c, such that the whole recurrence relation holds for $T(n) \leq cnlog^{(\frac{3}{4})}$

So we will write,

$$T(n) \leq cn^{(log_3 4)} \leq cn^{log_3 4}$$

So, we need to know a value of c for which $n^{log_3 4}$ will overshadow n, so we can write

$$cn^{log_3 4} \geq n$$
$$c \geq n^{1-log_3 4}$$

Here we can assume that $1 - log_3^4 \leq 0$, since $log_3^4 > 1$

So by choosing a value of c, which ios sufficiently greater, the inequality holds

$$1 - log_3^4 \leq 0$$
$$log_3^4 \geq 1$$

Therefore proved.

4. (20 points)

Solution:

Psuedocode

```
Sort(A,n) :
        if n ≤ 1 :
                return
        Sort(A, n-1)
        for 2 to n:
                key = A[n]
                i = n-1
```

```
        while i > 0 and A[i] > key:
            A[i+1] = A[i]
            i-=1
        A[i+1] = key
    return
```

To calculate the recurrence relation for the worst case,

$$T(n) = T(n-1) + c(n-1) + d$$

By using iteration method,

1st Iteration :

$$T(n) = T(n-1) + c(n-1) + d$$

2nd Iteration :

$$T(n) = T(n-2) + c(n-2) + d + c(n-1) + d$$

3rd Iteration :

$$T(n) = T(n-3) + c(n-3) + dc(n-2) + d + c(n-1) + d+$$

(n-1)th Iteration :

$$T(n) = T(1) + c(n-(n-1)) + c(n-(n-2)) + ... + (n-1).d$$

$$T(n) = T(1) + c(1+2+3+4.. + (n-1)) + ... + (n-1).d$$

$$T(n) = T(1) + c(\frac{n(n-1)}{2}) + (n-1).d$$

$$T(n) = k + c(\frac{n(n-1)}{2}) + (nd-d)$$

$$T(n) = k + \frac{cn^2}{2} - \frac{cn}{2} + (nd-d)$$

We combine, k, $-\frac{cn}{2}$, -d to a single term to parse the logic better

$$T(n) = \frac{cn^2}{2} + nd + a$$

where a = $k - \frac{cn}{2} - d$

Therefore, the expression we can understand that the term n^2, which increases faster than the other linear terms. Thus proving that $T(n) = O(n^2)$

5. (20 points)

Solution:

Here it is given that,

$$max(f(n), g(n)) = \theta(f(n) + g(n))$$

Since θ is the tight bound
We can say that, we need to prove

$$max(f(n), g(n)) = \Omega(f(n) + g(n))$$

$$max(f(n), g(n)) = O(f(n) + g(n))$$

We can also write that as,

$$0 \leq c_1(f(n) + g(n)) \leq max(f(n), g(n)) \leq c_2(f(n) + g(n))$$

Here the functions are asympotitically non negative, So for some $n_0 > 0$, $f(n) \geq 0$, $g(n) \geq 0$, there will exist

$n \geq n_0$
So we can write,

$$f(n) + g(n) \geq max(f(n), g(n))$$

Here, we know that,

$$f(n) \leq max(f(n), g(n))$$
$$g(n) \leq max(f(n), g(n))$$

So,

$f(n) + g(n) \leq max(f(n), g(n)) + max(f(n), g(n))$

$f(n) + g(n) \leq 2.max(f(n), g(n))$

$\frac{1}{2}(f(n) + g(n)) \leq max(f(n), g(n))$

So here we can write,

$$0 \leq \frac{1}{2}(f(n) + g(n)) \leq max(f(n), g(n)) \leq (f(n) + g(n)) for n \geq n_0$$

So we can say that

$$(f(n), g(n)) = \theta(f(n) + g(n))$$

there will exist $c_1 = \frac{1}{2}$, $c_2 = 1$

6. (20 points)

Solution:

To prove that $2^{(n+1)} = O(2^n)$
Here, we find constants c and n_0, such that $\epsilon\ n \geq n_0$

$$2^{n+1} \leq c.2^n$$

$$2^n.2 \leq c.2^n$$

Lets assume c=3 and $n_0 = 1$, so we can say

$$2 \leq 3$$

The above inequality holds true for all $n \geq 1$ and $c \geq 3$. Thus the condition for O is satisfied.
To prove $2^2n = O(2^n)$
We need to prove that, there are positive constants, c and n_0 such that $\epsilon\ n \geq n_0$, $f(n) \leq c.g(n)$
So we can write

$$2^2n \leq c.2^n$$

Dividing by 2^2n

$$\frac{2^2n}{2^2n} \leq \frac{c.2^n}{2^2n}$$

$$1 \leq \frac{c.2^n}{2^2n}$$

$$1 \leq \frac{c}{2^n}$$

$$2^n \leq c$$

Here,

$$n \leq log_2c$$

This condition holds true if and only if $n \leq log_2c$

Since, 2^2n grows exponentially, and 2^n grows linearly according to n, we cannot say that 2^2n will have $O(2^n)$

So, $2^{2n}! = O(2^n)$