Name: Roll No:

National Institute of Technology Calicut
Department of Computer Science and Engineering
CS2005 Data Structures and Algorithms
First Midterm Examination, January 2014

Time: 1 Hour Maximum Marks : 20

20

(Note: For all the questions given below write your answers only in the space provided in the question paper. Answers written elsewhere will not be evaluated)

1. Solve the recurrences given below. Assume T(1)=1.

   a) $T(n) = 3T(n/3) + \Theta(n^2)$. Applying Master's theorem: 1Mark

Here $a = 3$, $b = 3$, $f(n) = \Theta(n^2)$ $n^{\log_b a} = n^{\log_3 3} = n^1$. Testing for conditions for case III (i) $3\left(\frac{n}{3}\right)^2 \leq \frac{3}{9}n^2 < 0.5\,n^2 \Rightarrow \exists c \;\&\; af\left(\frac{n}{b}\right) < cf(n)$

(ii) $f(n) = \Theta(n^2) = \Omega(n^{1+0.5})$

Hence Case III holds: So $T(n) = \Theta(n^2)$

   b) $T(n) = 2T(4n/6) + \Theta(n^2)$ Applying Master's theorem 1Mark

Here $a = 2$ $b = 6/4 = 1.5$ $f(n) = \Theta(n^2)$. Testing for Case III

$n^{\log_b a} = n^{\log_{1.5} 2} = n^{1.7}$. Condition I: $af\left(\frac{n}{b}\right) = 2\left(\frac{n}{1.5}\right)^2 = \frac{2}{2.25}n^2 < 0.9n^2$

Condition 2: $n^2 = \Omega(n^{1.7+0.2})$ $\epsilon = 0.2$. Hence Case III holds.

So $T(n) = \Theta(n^2)$

   d) $T(n) = T(n/2) + \Theta(1)$. 1 Mark

Applying Master's theorem

$a = 1$ $b = 2$ $f(n) = \Theta(1) = c \cdot n^0$

$n^{\log_b a} = n^{\log_2 1} = n^0$ $\therefore f(n) = \Theta(n^{\log_b a}) = \Theta(n^0)$.

Case II holds: So $T(n) = \Theta(n^0 \log_b n) = \Theta(\log n)$

2. Which of the following statements regarding Merge sort is not correct? 1 Mark

   a) The recurrence for Merge sort algorithm is $T(n)= 2(T(n/2)) +\theta(1)$

   b) Merge sorts runs in $\theta(n\log n)$ time in the best case.

   c) Merge sort runs in $O(n^2)$ time in the worst case.

   d) The space complexity of the Merge operation is $\theta(n)$.

   Ans: a.

3. Which of the following arrays is not a Max-heap? 1 Mark

a) 12 8 6 3 4 1 **b)** 21 20 19 18 17 16 **c)** 15 11 13 10 9 12 **d)** 56 50 42 48 46 44

Ans: d

4. State True/False for the following statements. 1 Mark

   a) Insertion sort runs in $\theta(n)$ time in the best case.

Ans: True

   b) Insertion sort runs in $\Omega(n)$ time in the worst case.

Ans: True

Name: Roll No:

5. What would be the minimum and maximum number of elements in a heap of height h? 1 Mark

Minimum: 2^h .... Maximum: $2^{h+1}-1$ ....

6. Write True/False for the following equality. Justify your answer.

a) $100n^2 + 30n + 1000 = O(n^3)$. True. We demonstrate $\exists c, n_0 : fn \le cn^3 \ \forall n > n_0$ 1Mark

$$100n^2 + 30n + 1000 \le 100n^2 + 30n^2 + 1000n^2 \text{ for } n \ge 2$$
$$\le 100n^3 + 30n^3 + 1000n^3 \text{ for } n \ge 2$$
$$\le 1130n^3 \text{ for } n \ge 2$$

As there exists c & n_0 $\le cn^3$ for $n \ge n_0$ for $c = 1130$ & $n_0 = 2$. $\exists c, \exists n_0$

Hence by definition $100n^2 + 30n + 100n$ is $O(n^3)$

7. Prove that $3n^2 = o(n^3)$ We prove by contradiction 1 Mark

Let the statement be false. i.e $3n^2 \ne o(n^3)$

Then $\exists c, c > 0 : 3n^2 < cn^3 \ \forall n \ge n_0$ is false for $\forall n_0$.

$\Rightarrow 3n^2 \ge cn^3$ for all n

$\Rightarrow 3 \ge cn$ for all n. Let $n = \frac{3}{c} + 1$

hen $\Rightarrow 3 > c(\frac{3}{c} + 1) \Rightarrow 3 > 3 + c \Rightarrow c < 0$ - contradiction

So $3n^2$ is $o(n^3)$

8. Solve the recurrence $T(n) = T(n^{0.25}) + \log_2 n$ 1Mark

Let $n = 2^m$

$T(2^m) = T(2^{0.25m}) + \log_2(2^m)$

i.e $T(2^m) = T(2^{m/4}) + m$ Let $T(2^m) = S(m)$.

$S(m) = S(m/4) + m$. Solving this by Master's theorem $m^{\log_4 1} = m^0 = 1$

$m = \Omega(m^{0+1})$. Also $\frac{m}{4} < 0.5m$. So Case III holds

$S(m) = \Theta(m)$

i.e $T(2^m) = \Theta(m) \Rightarrow T(n) = \Theta(\log_2 n)$

Prove or disprove.

9. For two positive functions f(n) and g(n) 1.5 Marks

If $f(n) = \theta(g(n))$ then $f(n) = \omega(g(n))$

We can show that the implication is false by showing that the assumption that the left hand side is true implies that the RHS is false.

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

$\Rightarrow (f(n) = O(g(n))) \wedge (f(n) = \Omega(g(n)))$

$\Rightarrow f(n) = O(g(n))$ (By definition $\exists n_0 \exists c \cdot c > 0 : f(n) \le c\,g(n) \ \forall n \ge n_0$)

$\Rightarrow f(n) \le c\,g(n) \ \forall n \ge n_0$

$\Rightarrow \neg(\forall c : f(n) > c\,g(n) \ \forall n \ge n_0)$

$\Rightarrow \neg(f(n)$ is $\omega(g(n)))$ by definition of $\omega(g(n))$

$\Rightarrow \neg(f(n) = \omega(g(n)))$

2.10 Iterative-Reverse (A)

1. $n = A.length$
2. for $i = 1$ to $\lfloor n/2 \rfloor$
3. Exchange $A[i]$ with $A[n-i+1]$

<u>Proof of Correctness</u>

On the input array $A = (a_1 \cdots a_n)$ we define the following terms.

Sequence (p, q) stands for the sequence $a_p\, a_{p+1} \cdots a_q$ if $p \leq q$ and the sequence $a_q\, a_{q-1}\, a_{q-2} \cdots a_p$ if $p > q$

i.e

$$\text{Sequence}(p, q) = \begin{cases} a_p \cdots a_q & , \text{ if } p < q \\ a_q \cdots a_p & , \text{ if } q < p \\ a_p & , \text{ if } q = p. \end{cases}$$

where a_i, a_j are the elements of the ~~the~~ input array at Indices i & j respectively.

<u>LOOP INVARIANT</u>

At the beginning of the iteration i, the array $A[1 \ldots i-1]$ contains the sequence Sequence$[n, n-i+2]$ and $A[n-i+2 \ldots n]$ contains the sequence Sequence$[i-1, 1]$ and the contents of $A[i \ldots n-i+1]$ are undisturbed.

Initialization

In the beginning $i = 1$

$A[1..0]$ is a null set, &

$A[n+1..n]$ is a null set — Hence trivially true,

$A[1...n]$ are undisturbed

Maintenance

If true at the beginning of i^{th} iteration then it is true at the beginning of $i+1^{th}$ iteration

If true at the beginning of i^{th} iteration

$$A[1..i-1] = \text{Sequence}[n, n-i+2]$$

and $A[n-i+2..n] = \text{Sequence}[i-1, 1]$

and $A[i...n-i+1]$ are undisturbed.

During the iteration, the only statement executed is exchange of $A[i]$ with $A[n-i+1]$

(Let $\bullet$ stand for concatenation)

Hence $A[1..i] = \text{Sequence}[n, n-i+2] \bullet A[n-i+1]$

$$= \text{Sequence}[n, n-i+1]$$

$$A[n-i+1..n] = A[i] \bullet \text{Sequence}[i-1, 1]$$

$$= \text{Sequence}[i, 1]$$

i.e $A[1...(i+1)-1] = \text{Sequence}[n, n-(i+1)+2]$

and $A[n-(i+1)+2..n] = \text{Sequence}[(i+1)-1, 1]$

Also the elements $A[i+1... n-i]$ are undisturbed

i.e $A[(i+1)... n-(i+1)+1]$ are undisturbed

$\therefore$ True for $i+1^{th}$ iteration.

Termination

(At the beginning of iteration $\left\lfloor \frac{n}{2} \right\rfloor + 1$)

(i) $A\left[1 \ldots \left\lfloor \frac{n}{2} \right\rfloor\right]$ contains ~~the~~ Sequence $\left(n, n - \left\lfloor \frac{n}{2} \right\rfloor + 1\right)$

(ii) $A\left[n - \left\lfloor \frac{n}{2} \right\rfloor + 1 \ldots n\right]$ contains Sequence $\left[\left\lfloor \frac{n}{2} \right\rfloor, 1\right]$

(iii) and elements $A\left[\left\lfloor \frac{n}{2} \right\rfloor + 1 \ldots n - \left\lfloor \frac{n}{2} \right\rfloor\right]$ are undisturbed.

Now, if n is even, then

$A\left[1 \ldots \frac{n}{2}\right]$ contains Sequence $\left(n, \frac{n}{2} + 1\right)$

$A\left[\frac{n}{2} + 1 \ldots n\right]$ contains Sequence $\left(\frac{n}{2}, 1\right)$

$A\left[\frac{n}{2} + 1 \ldots \frac{n}{2}\right]$ is a null set (undisturbed trivially).

So the array is perfectly reversed.

if n is odd, then let $k = \left\lfloor \frac{n}{2} \right\rfloor$ i.e $n = 2k+1$

$A[1 \ldots k]$ contains Sequence $(n, n-k+1)$

$A[n-k+1 \ldots n] = A[k+2 \ldots n] =$ Sequence $(k \ldots 1)$

and $A[k+1 \ldots k+1]$ is left undisturbed (middle element)

Hence the array is perfectly reversed.

Q11

```
Algo-solution (A, B)
    Heapsort (A)
    Heapsort (B)
    count = 0
    a = A.length
    b = B.length
    while (a ≥ 1) and (b ≥ 1)
        if A[a] > B[b]
            a = a-1
        else if A[a] = B[b]
            a = a-1
            b = b-1
            count = count + 1
        else b = b-1
return (count)
```

Space Complexity : Heap sorts incurs ~~$O(n)$~~ $O(\log n)$ recursions at the same time. ~~At~~ The remaining steps in the Algorithms use $O(1)$ space.

Time Complexity:

For Heapsort: $O(n \log n)$.

~~For~~ The while loop is executed at the most $(m+n)$ times

as $m = O(n)$.

Time complexity of while loop is $O(n)$

Hence Time Complexity $= O(n \log n)$