CS201: Data Structures

ANSWER ALL QUESTIONS

02-03-I | Time: 120 minutes

Full Marks: 60 | Date: 18-9-2002

1. Use idea of limits and apply L'Hospital's rule to prove the following [5]

$$f(n) + g(n) = O(n^2) \text{ where } f = 3n^2 - n + 4 \text{ and } g(n) = n \log n + 5$$

Solution (i): Consider $\lim_{n\to\infty} \frac{n \log n + 5}{3n^2 - n + 4}$. Applying L'Hospital's rule,

$$\begin{aligned}
\lim_{n\to\infty} \frac{n \log n + 5}{3n^2 - n + 4} &= \lim_{n\to\infty} \frac{\log n + 1}{6n - 1} \\
&= \lim_{n\to\infty} \frac{\log n}{6n - 1} \\
&= \lim_{n\to\infty} \frac{1}{6n} \\
&= 0
\end{aligned}$$

Therefore, $f(n)$ dominates $g(n)$ as n becomes large. Hence $g(n) = O(f(n))$. Implying $f(n) + g(n) = O(n^2)$

2. Apply the concept of limits to prove that any positive polynomial funtion grows faster than any polylogarithmic function. [6]

Solution: It is known that

$$\lim_{n\to\infty} \frac{n^b}{a^n} = 0$$

for all real constants a and b. Therefore, polynomial and exponentials are related as $n^b = o(a^n)$. A function $f(n)$ is *polylogarithmically* bounded if $f(n) = \log^{O(1)} n$. We can relate the growth rates of polynomial and ploylogarithms by substituting

1) $\log n$ for n, and
2) 2^a for a.

So we get

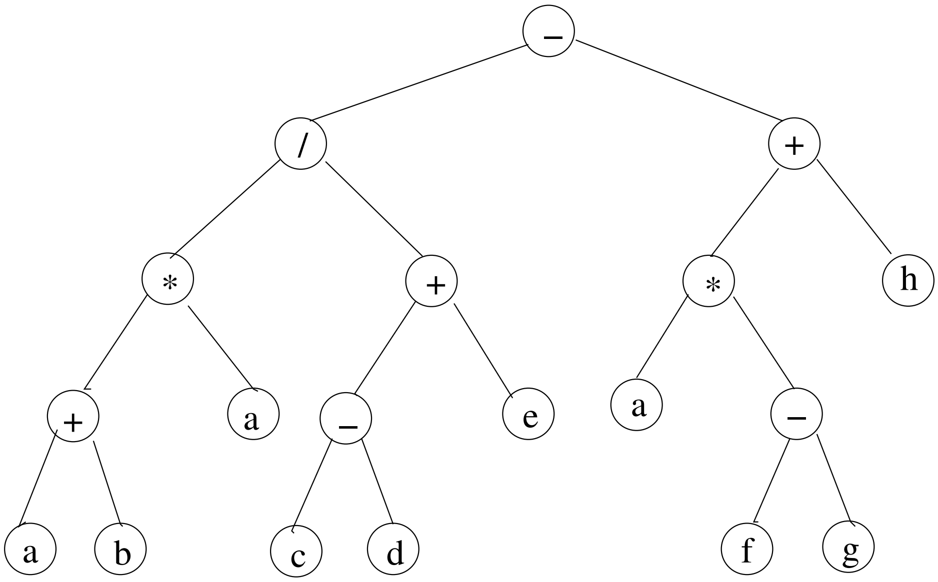
$$\lim_{n\to\infty} \frac{\log^b n}{2^{a^{\log n}}} = \lim_{n\to\infty} \frac{\log^b n}{n^a} = 0$$

3. Draw the expression tree representing the following infix arithmetic expression

$$(((a + b) * a)/(c - d + e)) - (a * (f - g) + h)$$

Refer to the tree to find the equivalent post-fix and prefix expressions. [11]

Solution: The expression tree is as follows:



Postfix form is: `ab+a*cd-e+/afg-*h+-`

Prefix form is: `-/*+aba+-cde+*a-fgh`

4. Draw a (single) binary tree T such that

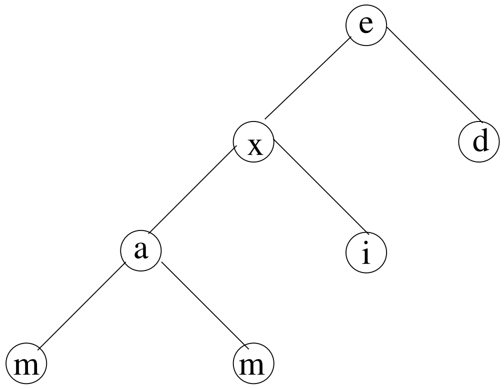
   (a) Each internal node stores a single character

   (b) A preorder traversal of T yields `e x a m m i d`

   (c) An inorder traversal of T yields `m a m x i e d`

[11]

Solution: The tree is as shown in the figure below:



5. Compute exact running time of the following code. Use the same notations and units for analysis of running time as discussed in the class. Also use the labels `L1, L2, L3, L4` to refer to the lines of code in your answer. [12]

```
     public class Example {
        public static int horner(int a[], int n, int x) {
L1          int result = a[n];
L2          for (int i=n-1; i>=0; --i)
L3          result = result*x+a[i];
L4          return result;
        }
     }
```

Solution:

Statement	*Running time*
L1	$3\tau_{fetch} + \tau_{[.]} + \tau_{store}$
L2a	$2\tau_{fetch} + \tau_{-} + \tau_{store}$
L2b	$(2\tau_{fetch} + \tau_{<}) \times (n+1)$
L2c	$(2\tau_{fetch} + \tau_{-} + \tau_{store}) \times n$
L3	$(5\tau_{fetch} + \tau_{[.]} + \tau_{+} + \tau_{\times} + \tau_{store}) \times n$
L4	$\tau_{fetch} + \tau_{return}$
Total	$(9\tau_{fetch} + 2\tau_{store} + \tau_{<} + \tau_{[.]} + \tau_{-}) \times n$ $+(8\tau_{fetch} + 2\tau_{store} + \tau_{<} + \tau_{[.]} + \tau_{-} + \tau_{return})$

6. Write an equivalent RAM prgram for the horner algorithm given in question 5. The values of `x` and `a[i]` for `i=n-1, n-2, ..., 0` can be read from the input tape. [15]

   Solution: The RAM program is

```
 label          Operation Code   Comment
-------        ----------------  ------------------
                READ    1        read x in r1
                LOAD   =v(n+1)   get n+1
                STORE   2        store n+1 in r2
                LOAD   =v(a[n])  get a[n]
                STORE   3        store result=a[n] in r3
Loop            LOAD    2
                SUB    =1
                JZERO   Over     Halt if n=0
                STORE   2        store n=n-1 in r2
                LOAD    1        load r0 = x
                MULT    3        find x*result
                READ    4        read a[n-1] in r4
                ADD     4        result = result+a[n-1]
                STORE   3        store result in r3
                JUMP    Loop     loop
Over            WRITE   3        write result
                HALT             Stop
```