

MIDTERM TEST 2

Introduction to Computer Science COMP 250

Mon. Nov. 14, 2016
Professor Michael Langer

Use the name that corresponds to your McGill email: FIRSTNAME.LASTNAME@mail.mcgill.ca

LAST NAME: _____

FIRST NAME : _____

McGILL ID: _____

GRADE: _____ /10

Instructions:

- This is a closed book exam. You are *not* allowed a crib sheet.
- No electronic devices. You may use your cell phone as a clock only.
- Turn over the exam when you are done (or leave, if you are close to a row exit.)
- Do not talk until *all* the exams are collected.

1. (2 points)

Use the method of back substitution to solve the following recurrence:

$$t(n) = 5n + t\left(\frac{n}{2}\right).$$

You may assume $n = 2^m$ and $t(1) = 0$. For full points, you must show your work.

Solution

$$\begin{aligned} t(n) &= 5n + t\left(\frac{n}{2}\right) \\ &= 5n + 5\frac{n}{2} + t\left(\frac{n}{4}\right) \end{aligned} \tag{1}$$

$$\begin{aligned} &= 5n + 5\frac{n}{2} + 5\frac{n}{4} + t\left(\frac{n}{8}\right) \\ &= 5n + 5\frac{n}{2} + 5\frac{n}{4} + 5\frac{n}{8} + \dots + 5\frac{n}{2^{m-1}} + t\left(\frac{n}{2^m}\right) \end{aligned} \tag{2}$$

$$\begin{aligned} &= 5\left(\sum_{i=1}^m 2^i\right) + t(1) \\ &= 10\left(\sum_{i=0}^{m-1} 2^i\right) \end{aligned} \tag{3}$$

$$\begin{aligned} &= 10(2^m - 1) \\ &= 10(n - 1) \end{aligned} \tag{4}$$

Grading

0.5 for each of (1-4).

We took off 0.5 for calculation error, if the steps were followed by gave the wrong answer.

We did not take off any points the sum in (3) was to m . This is easy-to-make off by one error. It leads to a different solution $t(n) = 5(2n - 1)$.

Another way to sum the geometric series is to factor out the n and compute:

$$\sum_{i=1}^{m-1} 2^{-i} = \frac{1 - \left(\frac{1}{2}\right)^m}{1 - \frac{1}{2}} = 2\left(1 - \frac{1}{n}\right)$$

2. (2 points)

- (a) Give the formal definition of “ $t(n)$ is $O(g(n))$ ”.

Solution

There exists c and n_0 such that for all $n \geq n_0$, $t(n) \leq cg(n)$.

Grading

0.5 points. Must use quantifiers correctly.

- (b) Use the formal definition of big O to show that if two functions $t_1(n)$ and $t_2(n)$ are both $O(n)$, then their product

$$t(n) = t_1(n) * t_2(n)$$

is $O(n^2)$.

Solution

There exists c_1 and n_1 such that, for all $n \geq n_1$, $t_1(n) < c_1 n$.

There exists c_2 and n_2 such that, for all $n \geq n_2$, $t_2(n) < c_2 n$.

Thus, if $n \geq \max(n_1, n_2)$, then

$$t_1(n) * t_2(n) < c_1 c_2 n^2$$

So take $c = c_1 c_2$, $n_0 = \max(n_1, n_2)$.

Grading

0.5 point for c . 0.5 for max condition. 0.5 for putting the argument together.

3. (2 points)

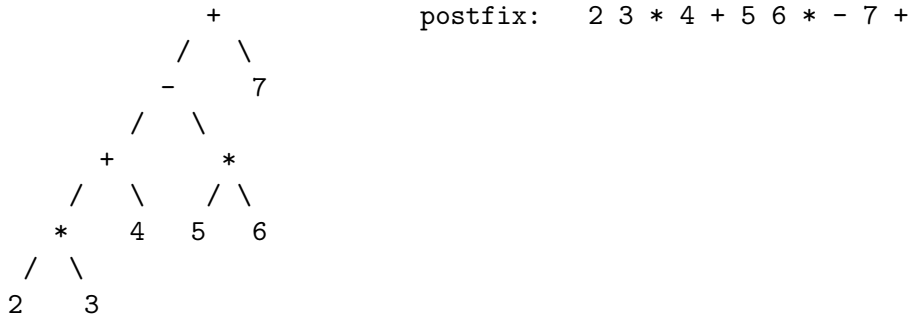
Consider the infix expression

$$2 * 3 + 4 - 5 * 6 + 7.$$

Assuming the usual precedence ordering for these operations, namely left before right, and $*$ before $+$ or $-$:

- (a) draw the corresponding “expression tree”,
- (b) rewrite the expression in postfix.

Solution



Grading

(a) 1 point for tree.

Gave only 0.5 if precedence wrong for the $+$, $-$ operations, or if subtrees were swapped (left vs. right).

(b) 1 point for postfix. If the tree you gave in (a) was incorrect, we gave the point for (b) if the postfix you gave was consistent with your tree from (a).

Did not give point if a unary operator (one argument) was used.

4. (2 point)

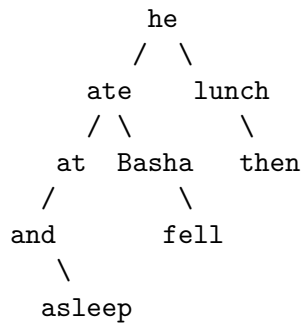
Form a binary search tree out of the words from the sentence:

he ate lunch at Basha and then fell asleep

The tree must be constructed by adding the words in the order they are found in the sentence, namely first add the word **he** to an empty tree. Then add the word **ate**, ...

Use the usual lexicographic ordering. Note that if s_1 is a prefix of s_2 , then $s_1 < s_2$.

Solution



Grading

Took off 0.5 for each mistake.

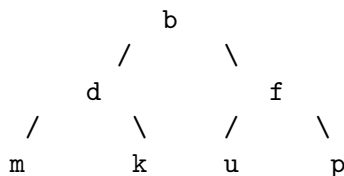
Gave 1 point if the tree was a BST but the placement of nodes did not follow the instruction "adding the words in the order they are found in the sentence".

5. (2 points)

Build a heap from the list of characters (m, d, p, b, k, u, f). Give your answer as a list, and as a tree. For full points, you must show your work.

Solution

1	2	3	4	5	6	7
(m	d	p	b	k	u	f)
(d	m	p	b	k	u	f)
(d	m	p	b	k	u	f)
(b	d	p	m	k	u	f)
(b	d	p	m	k	u	f)
(b	d	p	m	k	u	f)
(b	d	f	m	k	u	p)



Grading

The question did not specify which algorithm to use to build a heap, so we gave the points even if you used something other than the two algorithms given in class (*as long as you explained how you did it*). For example, you might have sorted the elements, which automatically gives a heap.

(b d f k m p u)

However, if the explanation was unclear, then we took off 0.5.

We took off 0.5 if there was no correspondence between list and tree (or one was missing but the other was correct).