QUIZ 4

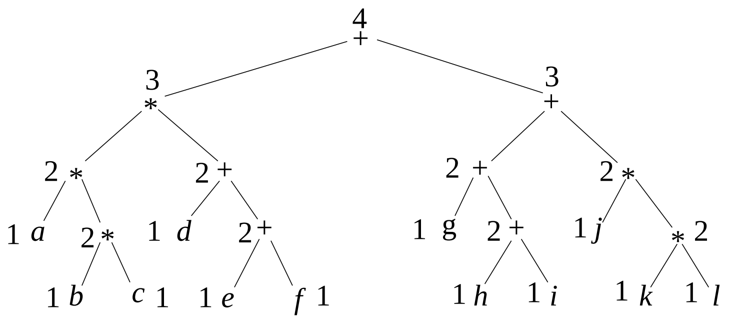
Date: April 12, 2002 Maximum marks: 35

1. Consider the expression $((a*(b*c))*(d+(e+f)))+((g+(h+i))+(j*(k*l)))$. Assume a machine with instructions:

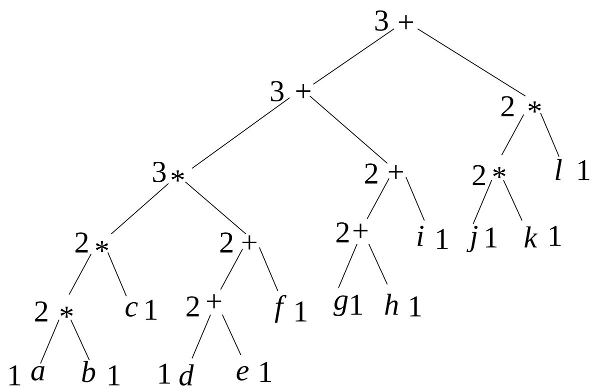
   $R_i \leftarrow R_i \; op \; R_j$
   $R_i \leftarrow m$
   $m \leftarrow R_i$

   where all instructions have a cost of 1. Note that the instruction $R_i \leftarrow R_i \; op \; m$ is not there.

   (a) Draw the expression in the form of a tree. Calculate the label at each node of the tree.



   (b) Using commutative and associative properties of the operators, rearrange the tree so that the label at the root is minimised. Once again label the tree.



   (c) Assuming that the machine has 2 general purpose registers R_1 and R_2, generate optimal code for the tree.

   $R_1 \leftarrow k$
   $R_2 \leftarrow j$
   $R_2 \leftarrow R_2 * R_1$
   $R_1 \leftarrow l$
   $R_2 \leftarrow R_2 * R_1$
   $m_1 \leftarrow R_2$

$R_1 \leftarrow h$
$R_2 \leftarrow g$
$R_2 \leftarrow R_2 + R_1$
$R_1 \leftarrow i$
$R_2 \leftarrow R_2 + R_1$
$m_2 \leftarrow R_2$

$R_1 \leftarrow e$
$R_2 \leftarrow d$
$R_2 \leftarrow R_2 + R_1$
$R_1 \leftarrow f$
$R_2 \leftarrow R_2 + R_1$
$m_3 \leftarrow R_2$

$R_1 \leftarrow b$
$R_2 \leftarrow a$
$R_2 \leftarrow R_2 * R_1$
$R_1 \leftarrow c$
$R_2 \leftarrow R_2 * R_1$

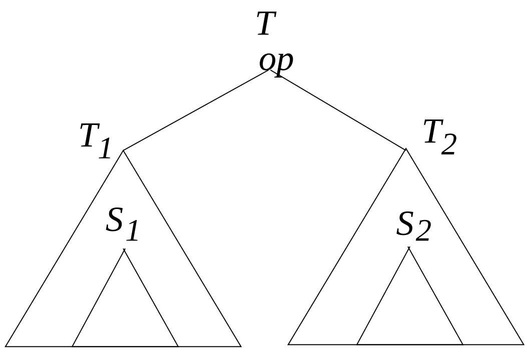
$R_1 \leftarrow m_3$
$R_2 \leftarrow R_2 * R_1$

$R_1 \leftarrow m_2$
$R_2 \leftarrow R_2 + R_1$

$R_1 \leftarrow m_1$
$R_2 \leftarrow R_2 + R_1$

2. Consider the following argument to show that the code generated by Sethi-Ullman is not necessarily in strong normal form.

   In the figure assume that stores are required at the roots of the subtrees S_1 and S_2. Also assume that the label of T_1 is the same as label of T_2.

   Then, parts of T will be evaluated as
   $S_2;\ store;\ T_2 - S_2;\ S_1;\ store;\ T_1 - S_1;\ op$

   This, clearly, is not in normal form. Is this argument right or wrong? Explain your answer.

The argument is fallacious. For this example, the root will be a major node and therefore a store is needed after evaluation of T_2.

T will be evaluated as:

$S_2;\ \ store;\ \ T_2 - S_2;\ \ store;\ \ S_1;\ \ store;\ \ T_1 - S_1;\ \ op$

This is indeed in normal form.

3. Consider a machine model with 2 general purpose registers and instructions shown below with their costs.

$R_i \leftarrow R_i\ op\ R_j$	cost – 2
$R \leftarrow c$	cost – 1
$R \leftarrow m$	cost – 1
$R \leftarrow ind(R)$	cost – 1
$R \leftarrow ind(R+m)$	cost – 5
$m \leftarrow R$	cost – 1

Now consider the expression $((a*b)*(ind(1+2)))*(ind(c+d))$. To start with, represent the expression in the usual tree form, traverse the tree in pre-order, and label the nodes with labels $L_0..L_{12}$.

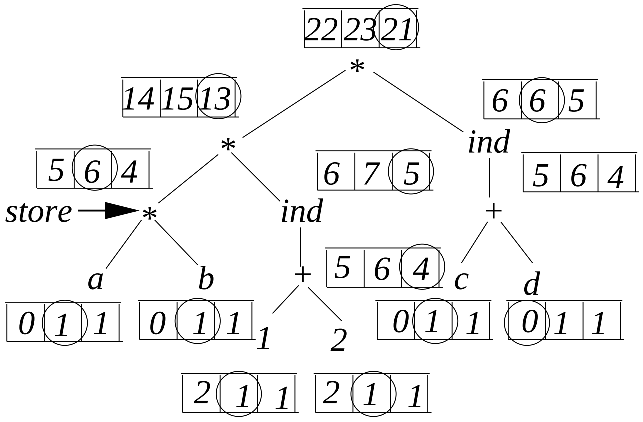
(a) Calculate the array of costs for each node in the tree.

(b) Encircle the least cost element at the root, and the element at each node which gave rise to the least cost element at the root.

(c) Indicate the permutation which gave rise to the encircled element at each node. For example if the encircled element at L_0 was achieved by evalating L_1 first and then L_9, write $L_0 \rightarrow L_1, L_9$. (Even if the choice of permutation is not important, arbitrarily select one).

(d) Indicate which nodes require a store.

(e) Generate optimal code for the expression. Label code fragments with labels from the expression tree.



The code generated is:

$R_1 \leftarrow a$ – code for the subtree to be stored
$R_2 \leftarrow b$
$R_1 \leftarrow R_1 * R_2$
$m \leftarrow R_1$

$R_1 \leftarrow 1$ – code for $ind(1+2)$
$R_2 \leftarrow 2$
$R_1 \leftarrow R_1 + R_2$

$R_1 \leftarrow ind(R_1)$

$R_2 \leftarrow m$ – code for $(a * b) * ind(1 + 2)$
$R_2 \leftarrow R_2 * R_1$

$R_1 \leftarrow c$ – code for $ind(c + d)$
$R_1 \leftarrow ind(R_1 + d)$

$R_2 \leftarrow R_2 * R_1$ – code for the root