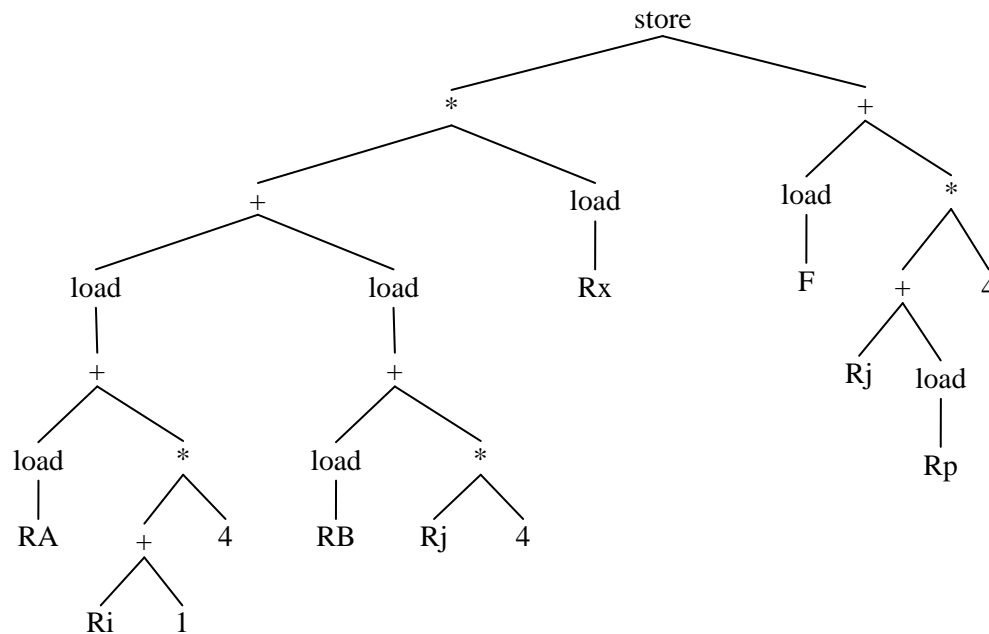


CS 6353 Compiler Construction, Homework #5

1. The following table shows the instruction set and the figure shows the operation tree generated for the statement $F[i+p] = (A[i+1] + B[j]) * x$.
- Title the tree using the maximal munching algorithm.
 - Title the tree using the dynamic programming algorithm.
 - Convert the results in (b) into machine instructions.

instruction	description	Cost
load addr, r1	load M[addr] to r1	20
store addr, r1	store M[addr] to r1	20
add r1, r2	$r1 := r1 + r2$	2
addc r1, c	$r1 := r1 + c$, where c is a constant	2
addm r1, addr	$r1 := r1 + M[addr]$	21
mul r1, r2	$r1 := r1 * r2$	8
mulc r1, c	$r1 := r1 * c$, where c is a constant	8
mulm r1, addr	$r1 := r1 * M[addr]$	27
move r1, r2, addr	$M[r1] := M[r2+M[addr]]$	62



2. Consider the following program. Assume that after executing 500 iterations of the loop in the program, garbage collection is invoked. How many bytes are freed?

```
pointer type is a pointer to a structure: { int value; pointer next }
global pointer first, last;
```

```
function add_link (val1, val2, index)
{ int val1, val2, index;

  pointer temp1 = new pointer;
  pointer temp2 = new pointer;
  temp1.value = val1; temp2.value = val2;
  if (index % 2 == 0) then
    { if (last != nil) then last.next = temp2;
      temp2.next = temp1; last = temp2;
    }
  else
    { if (last != nil) then last.next = temp1;
      temp1.next = temp2; last = temp1;
    }
}
```

```
main ()
{ int i, val1, val2;

  first = nil; last = nil;
  for (i=0; i<1000; i++) do
    { read val1, val2;
      add_link (val1, val2, 0);
    }
}
```