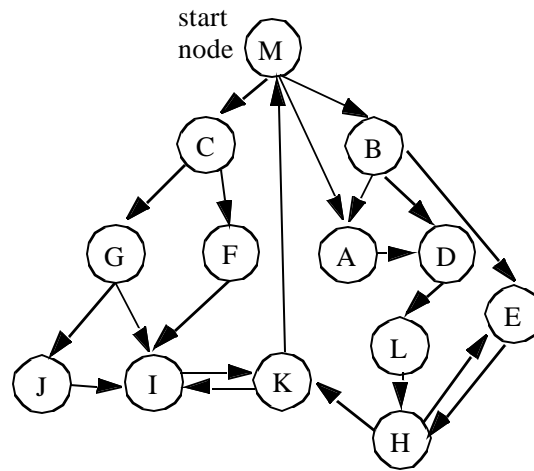
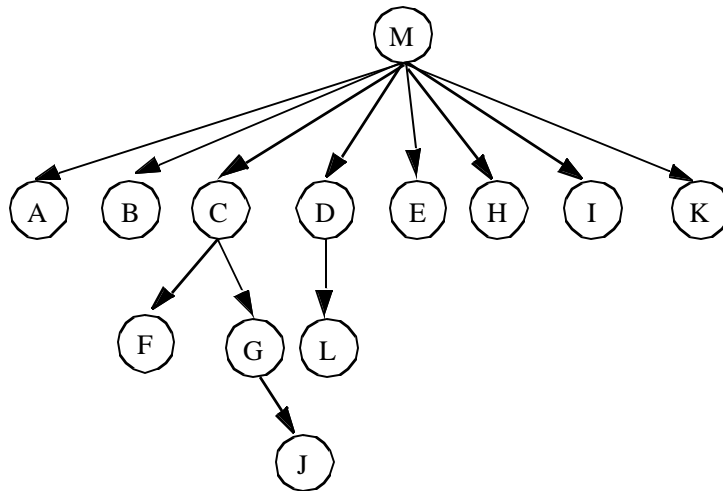




2. (20 points) Consider the following control flow graph:



And here is its dominator tree.



What are the natural loops in this flow graph? (Show intermediate steps for partial credit.)



4. (20 points) Consider the following program fragment:

```
int i, t, a[10000], b[10000]
for (i = 0; i < n; i++) {
    t = b[i];
    a[i] = t + t;
}
/* t is not used after this point */
```

It is obvious that we can rewrite the code as:

```
for (i = 0; i < n; i++) {
    int t;
    t = b[i];
    a[i] = t + t;
}
```

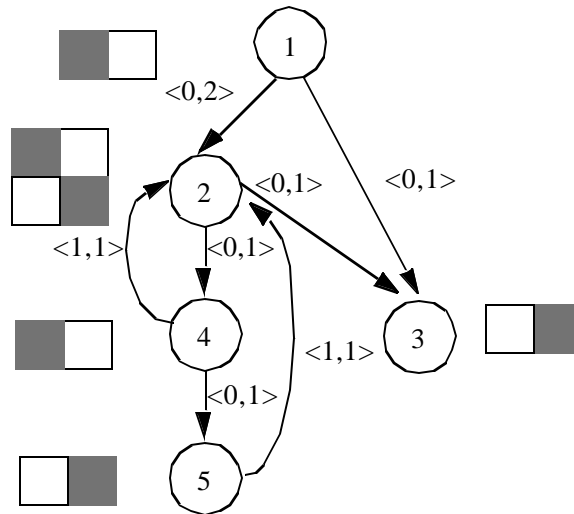
This transformation of shrinking the scope of a scalar variable to that of a loop body is known as *scalar privatization*. Giving each iteration a private version of the variable enables the iterations in this loop to be parallelized.

a. When is it legal to privatize a variable?

b. Describe an algorithm to detect when a scalar variable in a loop is privatizable.



6. (20 points) What is the best software pipelined schedule that you can create for the following precedence graph. This machine has two resources, R0, R1. The edges are labeled by the  $\langle \text{iteration difference}, \text{latency} \rangle$ ; each node is represented by its respective resource reservation table. If an instruction uses resource  $R_j$  in cycle  $i$  after it has been issued, ( $i = 0, 1$ ), it is indicated by a black square in row  $i$  and column  $j$  in the resource reservation table. For example, node 4 uses R0 in cycle 0, and R1 in cycle 1 after the node is issued.



- a. What is the bound of the initiation interval?
  
- b. Find the best software pipelined schedule. What is the initiation interval of your schedule? Show the schedule of one iteration.

7. (40 points) Consider the following program:

```

m = 100;
k = 0;
FOR i = 2 TO (m-1) {
  p = k;
  r = 20;
  FOR j = 2 TO (m-1) {
    if (B[i,j] > C[i,j]) {
      A[p] = A[r + 40*j];
    }
    E[i,j] = D[i,q];
    q = q+2;
    D[i,q-1] = E[i,j];
    p = p+1;
    r = r+1;
  }
}

```

You may use any of the techniques discussed in this course to answer the question, but you must specify the analyses you use to compute the answer.

a. Is the outermost loop parallelizable? Why? Explain how you get your answer; in particular, specify the data dependence tests performed.

b. Is the innermost loop parallelizable? Why? Explain how you get your answer; in particular, specify the data dependence tests performed.