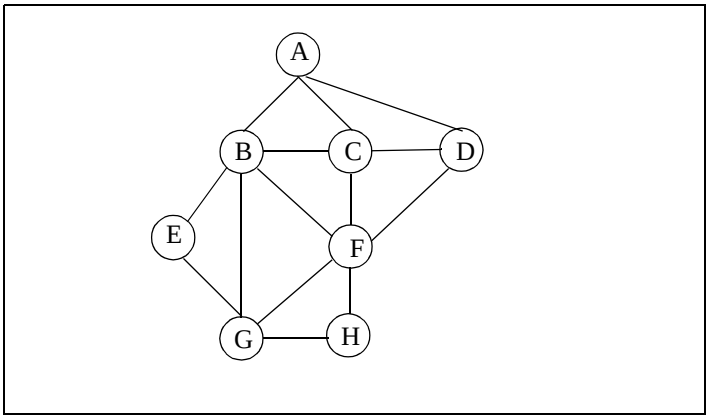
Sample Final Examination

1. (20 points) Consider the following interference graph:



   a. Give a machine with 3 registers, is it possible to find an allocation for the graph above without spilling? You may answer one of yes, no, “I don’t know”. If yes, give a coloring.

   b. How does the “improved Chaitin’s algorithm” behave when given this input? Recall that the improved Chaitin’s algorithm keeps variables that may need to be spilled on the “stack”, and tries to assign a register to them in the assignment phase.

2. (30 points) Suppose you have carefully defined a forward data flow algorithm for a framework that is distributive and has only finite descending chains.

   a. A colleague of yours accidentally deletes the code that initializes the entry node. Without telling you, he initializes the entry node to $\perp$ (bottom). Does that change the result of the analysis? Why?

   b. Suppose instead your colleague initializes all the *internal* nodes to $\perp$ (bottom).

      i. Will this algorithm give a *safe* answer for *all* flow graphs?

      ii. Will this new algorithm give a *safe* answer for *some* flow graphs? If so, which ones?

      iii. Will this new algorithm give the *meet-over-paths* answer for *all* flow graphs?

      iv. Will this new algorithm give the *meet-over-paths* answer for *some* flow graphs? If so, which ones?

3. (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:

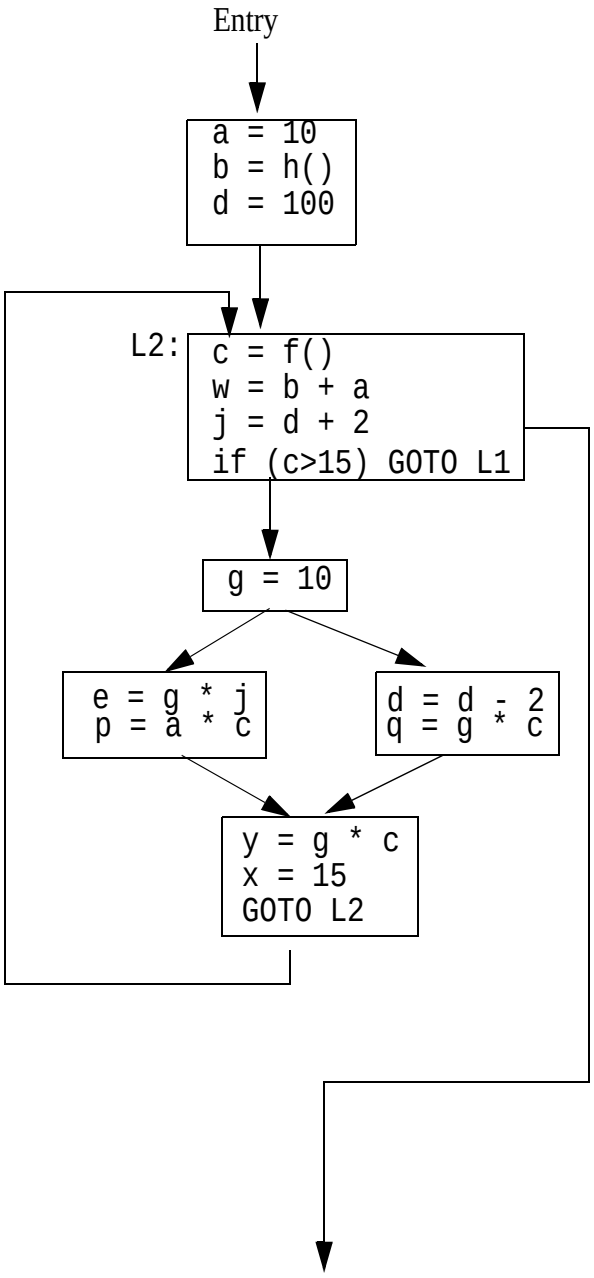
```
for (i = 0; i<n; i++) {
     int t;
     t = b[i];
     a[i] = t + t;
}
```

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?

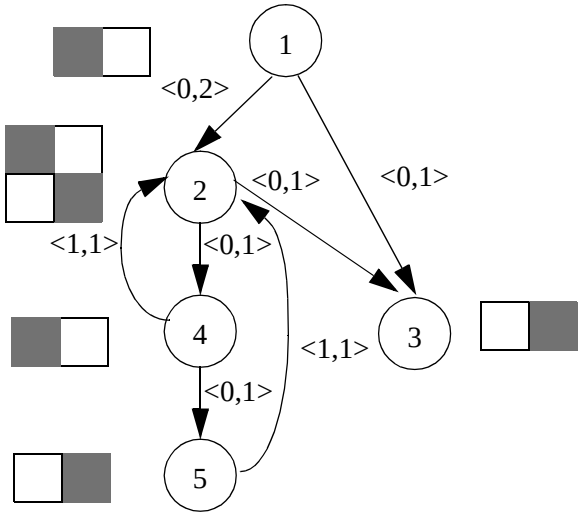
b. Describe an algorithm to detect when a scalabe variable in a loop is privatizable.

4. (30 points) Choosing from the machine-independent optimizations you learned in this class, apply the applicable ones to the following program. You should assume all the variables in the program to be live on exit. Clearly describe the sequence of optimizations you apply and show the final result.



5. (20 points) Consider the flow graph in Problem 4. What are the natural loops in this flow graph? (Show intermediate steps for partial credit.)

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 <iteration difference, latency>; each node is represented by its respective resource reservation table. If an instruction uses resource Rj 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.