CS243 Final Examination

Winter 2002-2003

You have 3 hours to work on this exam. The examination has 180 points. Please budget your time accordingly.

Write your answers in the space provided on the exam. If you use additional scratch paper, please turn that in as well.

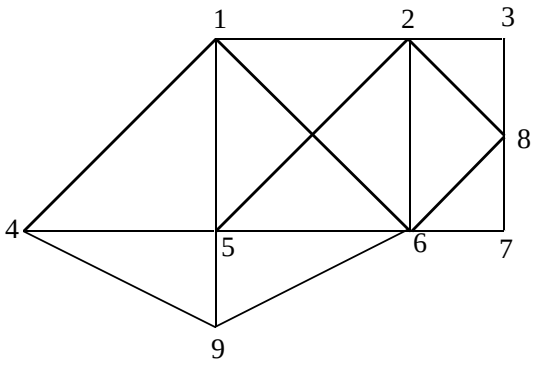
Your Name: _____

In accordance with both the letter and spirit of the Honor Code, I have neither given nor received assistance on this examination.

Signature: _____

Problem	Points	Score
1	10	
2	20	
3	30	
4	20	
5	30	
6	30	
7	40	
Total	180	

1. (10 points) Suppose you have a machine with 3 registers. Can you find a register allocation for the following interference graph such that no spilling is necessary? Your answer may be one of: yes, no, "I don't know". If the answer is yes, specify a register assignment. If the answer is no, explain why.



2. (20 points) Briefly give one important advantage and disadvantage for each of the following styles of garbage collection.

   a. Reference counting
      Advantage:

      Disadvantage:

   b. Copying garbage collection
      Advantage:

      Disadvantage:

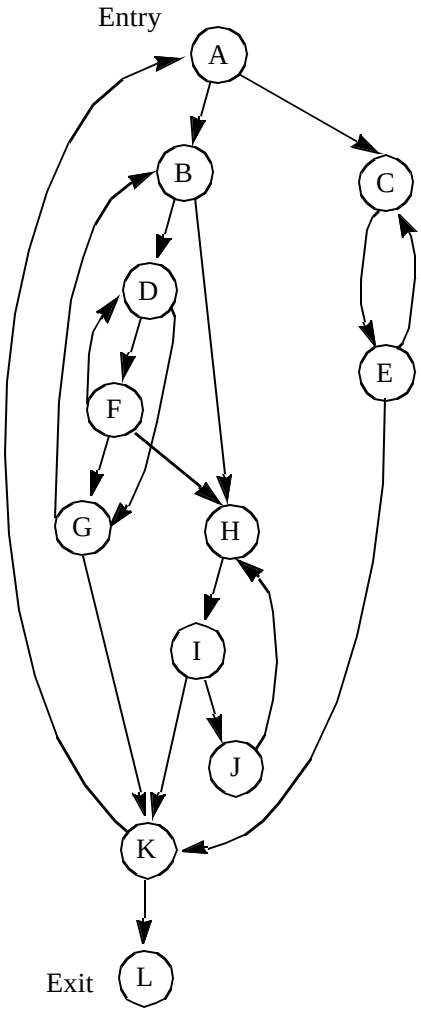
   c. Generational garbage collection
      Advantage:

      Disadvantage:

   d. Incremental garbage collection
      Advantage:

      Disadvantage:

3. (30 points) Consider the following control flow graph:



a. Give a reverse post-ordering for this flow graph.

b. Draw the dominator tree for this graph.

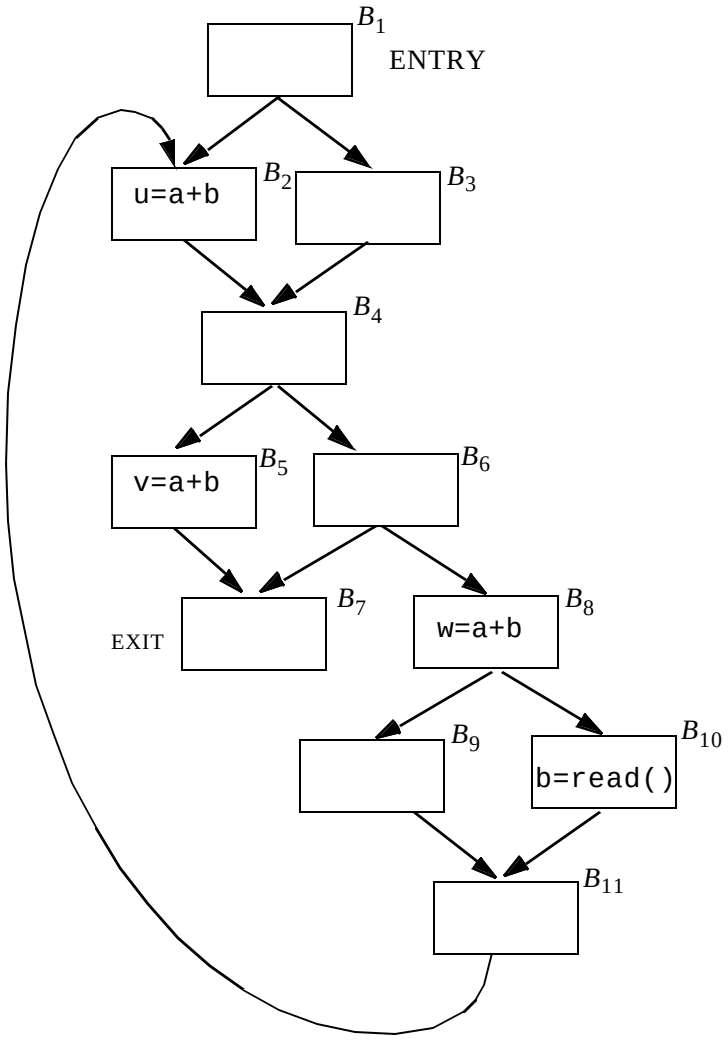
c. Find the natural loops in this graph.

d. Is the flow graph reducible? Explain.

e. Consider a data-flow solver that sweeps through all the nodes in reverse post-order until there are no more changes. Assume that the solver needs to iterate through the nodes once after the final answer is reached to detect convergence. If we apply the solver to the data-flow problem of finding dominators, what is the maximum number of times this solver has to iterate before it converges for the above program. Justify your answer.

f. What is the maximum number of times a reverse post-order solver has to iterate to find reaching definitions in this program? To justify the answer to part (e), show an example that requires the maximum number of iterations. You may do so by placing one definition in the program and identifying a program point whose value is not correct until the last but one iteration.

4. (20 points) Show the result of applying lazy code motion to the following program. Show the result of the optimization -- it is not necessary to show any intermediate steps.
(Introduce new basic blocks as necessary).



*** Extra page for answers ***

5. (30 points) Design a *monotone* iterative data flow analysis algorithm to find the signs of variables in a program. Your algorithm must determine, for each basic block, if each variable in the program is positive, non-positive, equal to zero, negative, non-negative, or unknown. The instructions in the program are: assignments, additions, subtractions, squares (square(x)=x*x), conditional and unconditional branches. You do not need to optimize the branch operations.

   a. What is the direction of your data flow analysis?

   b. Draw a diagram of the semi-lattice for one variable, identifying the top and bottom elements clearly.

   c. How do you initialize the iterative algorithm?

   d. Define the transfer function of a basic block.

   e. Is your transfer function distributive?

   f. Will your algorithm converge? If so, why?

6. (30 points) Consider an architecture that can issue one memory operation (LOD, STR), one arithmetic operation (ADD, MUL), and one control flow operation simultaneously. Addresses are of the form A[r+c], where r is a register, A is a base address, and c is a constant. LOD, STR, and ADD have latency 1; MUL has latency 2 (and is perfectly pipelined). All registers that are read in a cycle are read at the beginning of the cycle; all registers that are written are written at the end of the clock cycle.

   a. The following code is handed to a scheduler:

```
(1) LOD r1 = A[r2+0]
(2) MUL r3 = r3 * r4
(3) ADD r3 = r4 + r1
(4) STR B[r2+0] = r3
```

   List all dependencies in the following code and classify them as true, anti-, or output dependencies.

   b. Consider the following code:

```
L:
 (1) LOD r2 = A[r1+0]
 (2) LOD r3 = B[r1+0]
 (3) MUL r3 = r3 * r3
 (4) ADD r2 = r2 + r3
 (5) STR C[r1+0] = r2
 (6) LOOP L, r1, 1, 1000
```

      i. Give a lower bound on the initiation interval for a software pipelined version of this code? Justify your answer by showing the modulo resource reservation table.

      ii. Show the steady state of the software pipelined code. Use modulo variable expansion if necessary. Assume as many registers as necessary are available.

*** Extra page for answers ***

7. (40 points) Consider a small programming language defined as follows:
   A program is a statement.
   A statement is a sequence of primitive or loop statements separated by semi-colons.
   Let *u*, *v*, *w* below represent variables.

   A primitive statement can be one of:

   a. constant assignments: *u* = <constant>

   b. copy operations: *u* = *v*

   c. additions: *u* = *v* + *w*, *u* = *v* + <constant>

   A loop statement:

   Loop <constant> times {
   <statement>
   }
   where the constant is a positive number.

   Devise a general region-based "constant propagation" algorithm that can, for example, find the constant value of variable *d* in the following program:

```
a = 1
b = 2
c = a + b;
Loop 10 times {
    b = b + 1;
    loop 100 times {
        a = a + c;
    }
    b = b + 4;
}
d = a + b
```

*** Extra page for answers ***