631 exam practice mid-term

This program is to be used in questions 1-3.

```
S0        if (z <= 1) goto L1
S1        x = 1
S2        if (z > 2) goto L2
S3        y = x+1
S4        goto L3
S5   L1   x = 2
S6   L2   z = x − 3
S7        x = 4
S8        z = x+7
S9   L3   print x,y,z
```

1. Control flow analysis

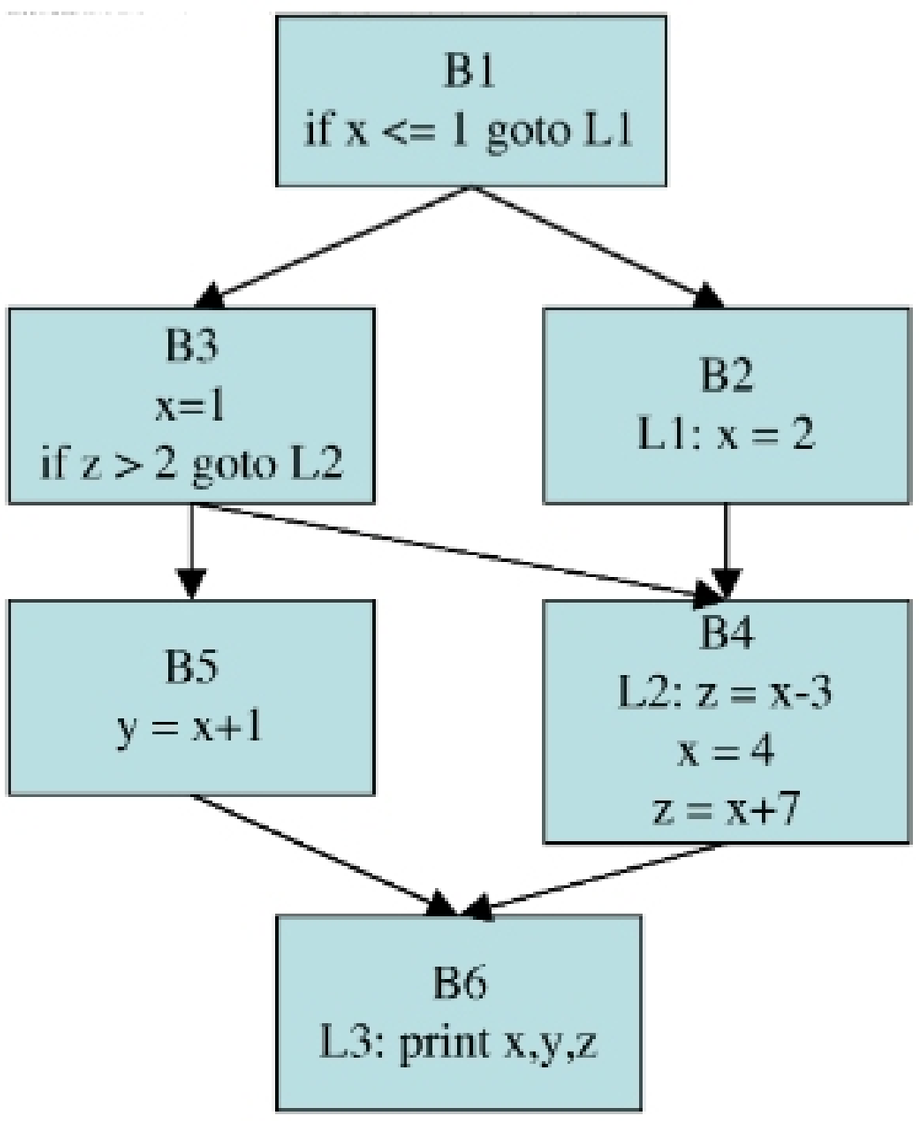
   (a) What are the basic blocks?

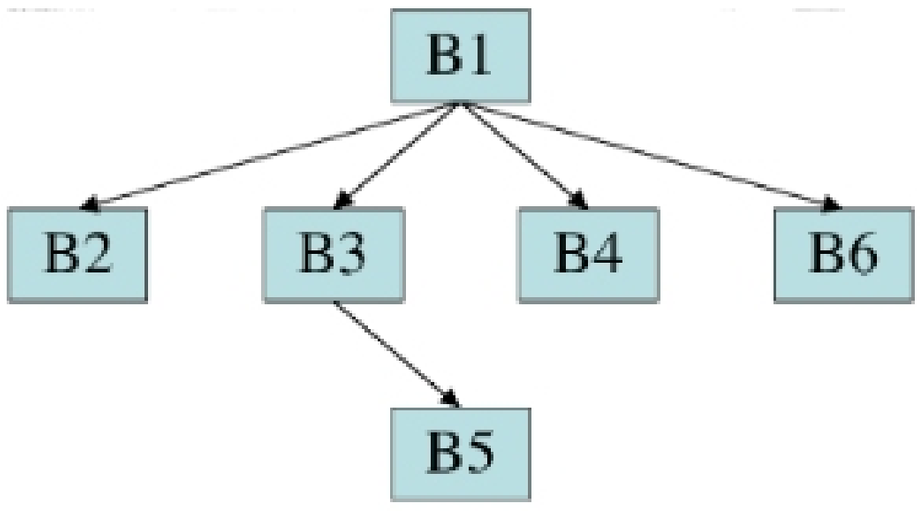
   (b) Show the control flow graph

   (c) Number the nodes of the CFG in reverse postorder.

   (d) Give the dominator tree for this program

Answer:





2. Control dependence - which block(s) are control-dependent on which block(s)?

Block	is control dependent on edge
B2	B1 → B2
B3	B1 → B3
B4	B3→B4 and B1 → B2
B5	B3 → B5

3. Dominance frontiers and SSA form

   (a) For each block, give the dominance frontier and the iterated dominance frontier.

Block	DF	DF^+
B2	B4	B4,B6
B3	B4,B6	B4,B6
B4	B6	B6
B5	B6	B6

   (b) Give the (minimal) SSA form of the program.

```
        if (z0 <= 1) goto L1
        x1 = 1
        if (z0 > 2) goto L2
        y1 = x1+1
        goto L3
L1      x2 = 2
L2      x4 = φ(x1, x2)
        z1 = x4 − 3
        x3 = 4
        z2 = x3+7
L3      x5 = φ(x1, x3)
        y2 = φ(y1, y0)
        z3 = φ(z0, z2)
        print x5,y2,z3
```

4. Data flow lattices

   (a) Remember that we define our $\sqsubseteq$ operator as $u \sqsubseteq v \equiv u = u \sqcap v$. A frame work is monotone if and only if $u \sqsubseteq v \Rightarrow f(u) \sqsubseteq f(v)$.

   Prove or disprove that even if a framework is not distributive (i.e., $f(u \sqcap v) = f(u) \sqcap f(v)$), monotonicity guarantees that $f(u \sqcap v) \sqsubseteq f(u) \sqcap f(v)$.

   Answer: First, we prove that $u \sqcap v \sqsubseteq v$. To prove this, we need to prove $u \sqcap v = u \sqcap v \sqcap v$, which is true since $\sqcap$ is associative $v \sqcap v = v$.

Similarly, $u \sqcap v \sqsubseteq u$.

To prove $f(u \sqcap v) \sqsubseteq f(u) \sqcap f(v)$, we need to prove that $f(u \sqcap v) = f(u \sqcap v) \sqcap f(u) \sqcap f(v)$.

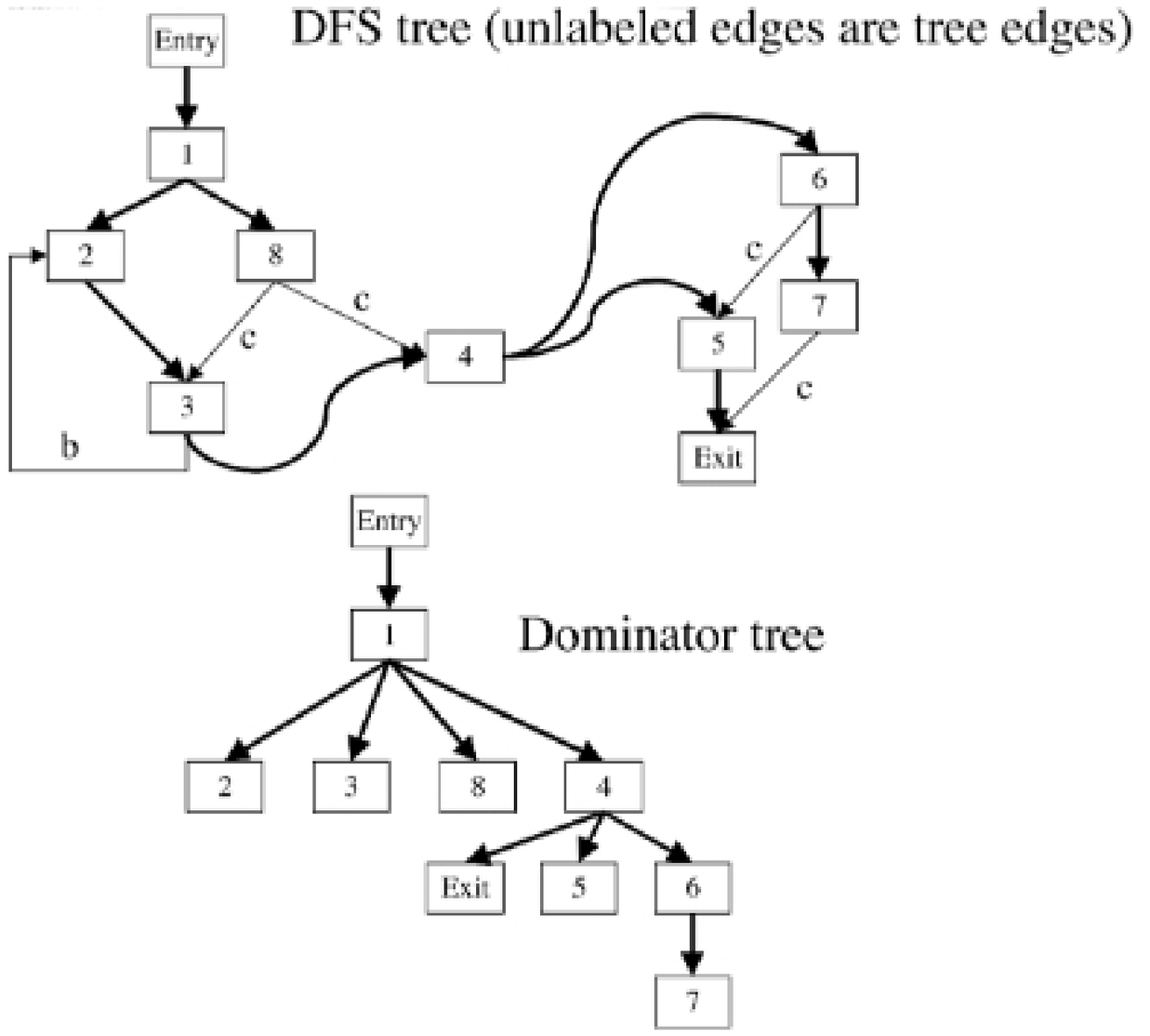
Since $u \sqcap v \sqsubseteq u$ and f is monotone, we have $f(u \sqcap v) \sqsubseteq f(u)$ and therefore $f(u \sqcap v) = f(u \sqcap v) \sqcap f(u)$. Therefore, our proof obligation is reduced to $f(u \sqcap v) = f(u \sqcap v) \sqcap f(v)$.

Similarly, we know $f(u \sqcap v) \sqsubseteq f(u)$ and use that to reduce our proof obligation to $f(u \sqcap v) = f(u \sqcap v)$, which is trivially true.

(b) Informally/intuitively, describe what would be true of a system that is not monotone.

Answer: Learning more would result in you knowing less. The most likely situation would be if the analysis decides that a certain path is unfeasible, it is tempting to reset the lattice value to $\top$; don't do this, it makes the framework non-monotone.

5. Depth First Search tree and dominator tree:



6. We want to be able to determine if a pointer might be null. We want to warn the user and refuse to compile when we find a statement that is a definite error, and omit run-time null pointer checks whenever possible.

Assume that doing a reference through a null pointer throws an exception that is not caught within the code we are analyzing.

For our analysis, try to compute accurate information for copy statements (e.g., after `p = q`, whatever we had previously known about `q` we now know about `p`) and pointer uses (e.g., after `p = q.x` we know that `q` is non-null and don't know anything about `p`).

Formulate this problem as one or more data flow problems. There are several different ways to do this; the more accurate your solution the higher your grade.

Answer: This is like a constant propagation problem, where the possible values are null or non-null. There are other ways to do it, but this is the way I will be doing it.