6.1100 Spring 2024 Miniquiz #6

Please submit your answers on Gradescope by April 11th, 2024, 11:59pm.

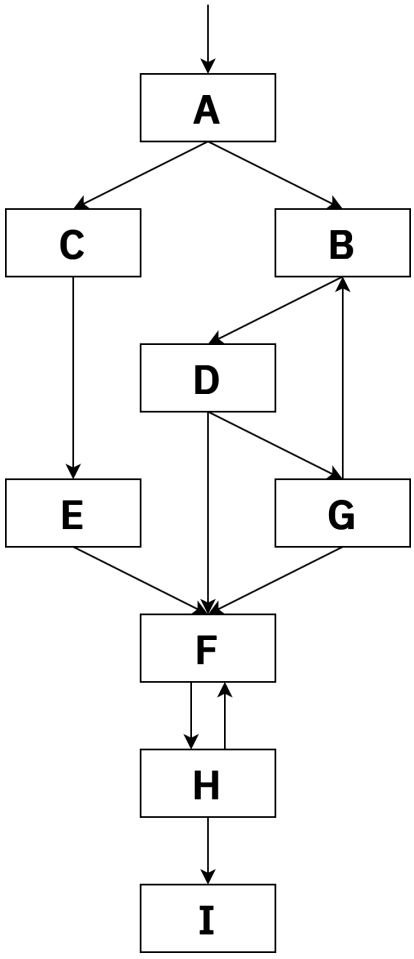
Name: 6.110 Staff

Email: 6.110-staff@mit.edu

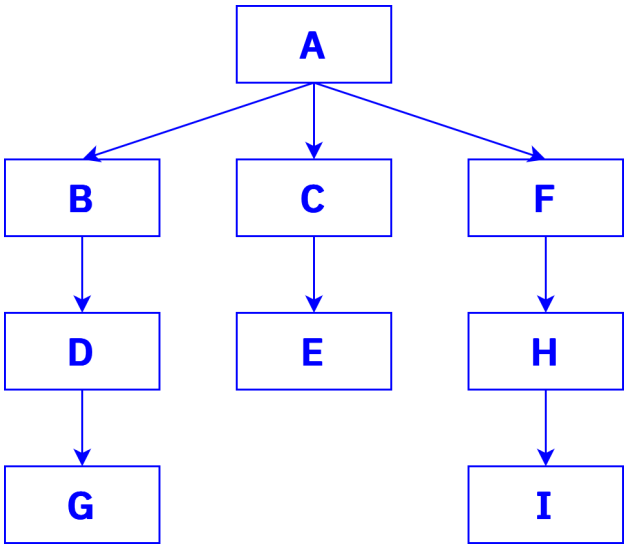
Note: This miniquiz covers loop optimizations and parallelization. We will cover register allocation in a later miniquiz.

1. Loop Optimizations

(For part a. and b.)
In the following control flow graph, **A** is the entry node and **I** is the exit node.



a. Draw the **dominator tree** of this control flow graph.



b. There are two loops in this control flow graph. Fill in the information about each loop in the table below.

Loop header	**Nodes in loop**
B	B, D, G
F	F, H

(For part c.-f.)

Consider the following program.

```
1  x = 0;
2  y = 5;
3  while (x < 10) {
4      x = x + 1;
5      a = y * y;
6      z = 8 * x + 24;
7      printf("%d %d\n", a, z);
8  }
```

c. Which line of code is **loop-invariant**? (Give the line number)

Line 5.

d. Which variable is the **base induction variable**?

x is the base induction variable.

e. Which variable is a **derived induction variable**?

z is a derived induction variable, with triple <x, 8, 24>.

f. Rewrite the program after **loop invariant code motion** and **induction variable optimizations** mentioned in class.

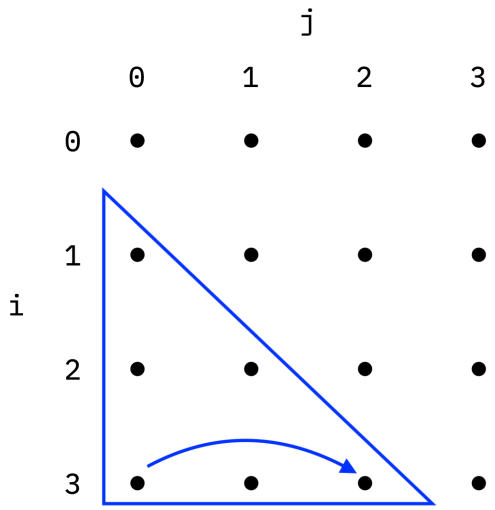
```
1  z = 24;
2  y = 5;
3  a = y * y;
4  while (z < 104) {
5      z = z + 8;
6      printf("%d %d\n", a, z);
7  }
```

2. Parallelization

Consider the following loops, where `A[i, j]` refers to the element in the i-th row and j-th column in a two-dimensional array.

```
for (i = 0; i < n; i += 1) {
    for (j = 0; j < i; j += 1) {
        A[i, j] = A[i, j - 2] + 3;
    }
}
```

a. Assume n = 4. In the grid below, circle the **iteration space** for the loops and draw the **distance vectors**. You may ignore out-of-range cases.



b. What is the **distance vector** for these loops?

dv = [0, 2]

c. Which of the loops (inner and/or outer) can be parallelized into a for-all loop?

Only the outer loop (the `i` loop) can be parallelized into a for-all loop.