MIT 6.035 Spring 2011 Quiz 3

Full Name: ____________________

MIT ID: ____________________

Athena ID: ____________________

Run LaTeX again to produce the table

1. Consider a simple machine with one ALU and one memory unit. The machine can execute an ALU operation and a memory operation at the same clock. A memory load operation takes two clocks (i.e. its result will be available in two clocks) and all other operations (including a memory store operation) complete in one clock.

   Consider the following code fragment:

```
L1:        mov 0(%rsp), %r10
L2:        mov %r10, 4(%rsp)
L3:        mov 8(%rsp), %r11
L4:        add %r8, %r11
L5:        add %r10, %r11
L6:        mov %r11, 12(%rsp)
```

   where `mov` *`src, dst`* copies a value from *`src`* to *`dst`* and `add` *`src, dst`* means *`dst`* `+=` *`src`*.

   (a) (5 points) A dependence DAG represents the constraints on the possible schedules of the instructions in a basic block. The nodes in a dependence DAG represent machine instructions and its edges represent dependencies between the instructions. Each edge is labeled with latency ℓ indicating that the *destination* node must be issued no earlier than ℓ clocks after the *source* node is issued. Draw the dependence DAG for the above code fragment.

(b) (8 points) How many clocks does it take to execute the code fragment without any instruction scheduling? Fill the following table with the instruction labels (`L1`, `L2`, $\cdots$ `L6`) that describes what instructions are executing in the memory unit and the ALU at each clock. Leave blank any clocks that are not needed.

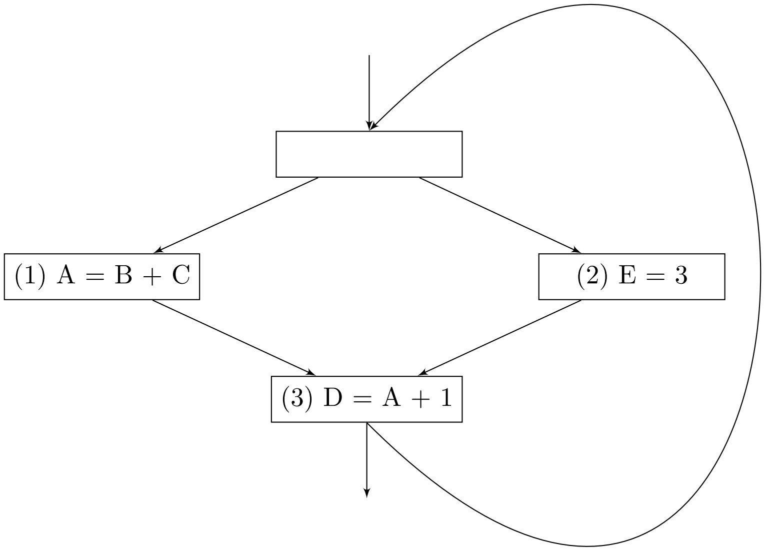
	Clock									
	1	2	3	4	5	6	7	8	9	10
Memory Unit										
ALU										

(c) (8 points) How are the instructions in the code fragment reordered when we apply the list-scheduling algorithm with the heuristics for selection presented in Lecture 13? Write your answer by listing the instruction labels (`L1`, `L2`, $\cdots$ `L6`) in the new order.

(d) (8 points) How many clocks does it take to execute the reordered code fragment after applying the list-scheduling algorithm? Fill the following table with the instruction labels (`L1`, `L2`, $\cdots$ `L6`) that describes what instructions are executing in the memory unit and the ALU at each clock. Leave blank any clocks that are not needed.

	Clock									
	1	2	3	4	5	6	7	8	9	10
Memory Unit										
ALU										

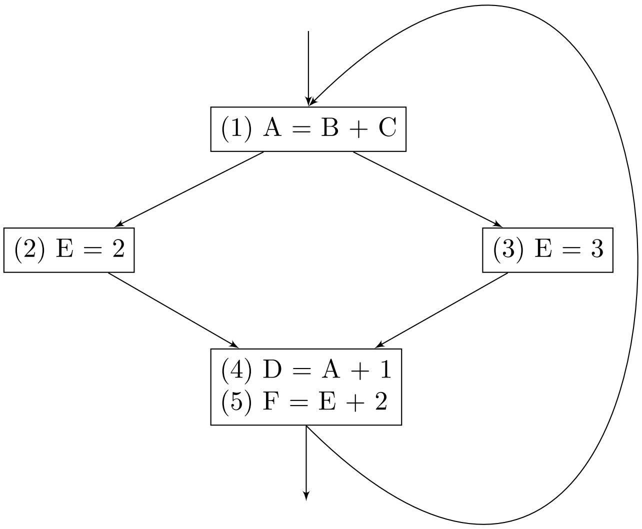
2. (a) Consider the following control flow graph with the assumption that all variables are live on exit from the loop:



i. (5 points) Which statements are loop-invariant?

ii. (5 points) Which statements can we move out of the loop?

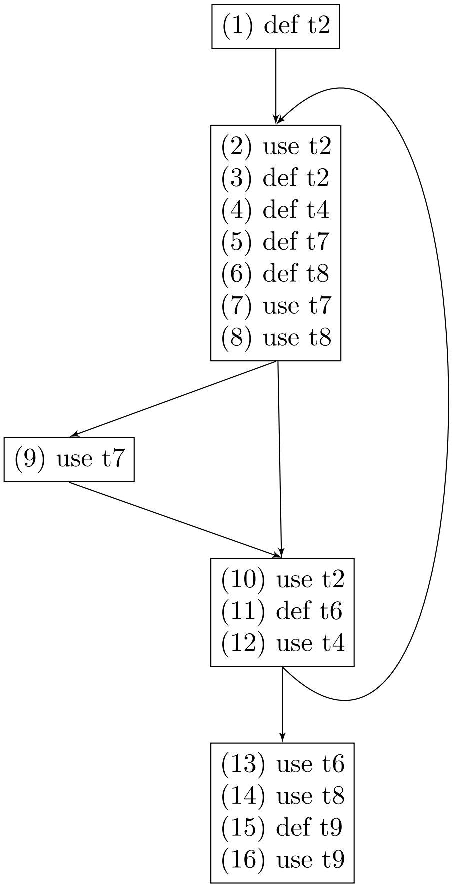
(b) Consider the following control flow graph with the assumption that all variables are live on exit from the loop.



i. (5 points) Which statements are loop-invariant?

ii. (5 points) Which statements can we move out of the loop?

3. Consider the following control flow graph:



(a) (8 points) Identify all the webs in the graph. Leave blank any rows or cells that are not needed.

Web Number	Variable	Instruction Number															
1																	
2																	
3																	
4																	
5																	
6																	
7																	
8																	
9																	
10																	
11																	
12																	
13																	
14																	

(b) (8 points) Draw the interference graph for the webs you identified in (a). Label each node with its web number and variable.

(c) (5 points) How many colors are required to color the nodes of the graph such that no two adjacent nodes share the same color?

(d) (5 points) The target machine has less registers than required to allocate a register to every web. Which web will be chosen to be spilled first when we adopt the interference-degree heuristic?

(e) (5 points) Which web will be chosen to be spilled first when we adopt the spill-cost heuristic?

4. (a)

```
FOR I = 1 TO N
    FOR J = 1 TO N
        A[I, J] = A[I - 1, J - 1] + 1;
    }
}
```

i. (3 points) What is the distance vector between the source and sink of a dependence in the iteration space of the loop nest?

ii. (3 points) What kinds of dependencies exist in the loop nest?

iii. (3 points) Which loop(s) can be *fully* parallelized?

(b)

```
FOR I = 1 TO N
    FOR J = 1 TO N
        A = A + A;
    }
}
```

i. (4 points) What is the distance vector between the source and sink of a dependence in the iteration space of the loop nest?

ii. (4 points) What kinds of dependencies exist in the loop nest?

iii. (3 points) Which loop(s) can be *fully* parallelized?