For the code below draw the constraint graph. Label each edge with the cause of the constraint (i.e. “RAW X2”, etc.).

```
LOOP:
  1: LDUR X1, [X6, #40]
  2: ADD  X5, X5, X1
  3: STUR X1, [X5, #20]
  4: ADDI X6, X6, #4
  5: SUBI X5, X5, #4
  6: CBZ  X5, LOOP
```

```
1: LDUR
   RAW X1 ←

2: ADD
   RAW X5

3: STUR

4: ADDI
   WAR X6

5: SUBI
   WAR X5

6: CBZ
   RAW X5

Control
```
For the code below draw the constraint graph. Then adjust the code to run as fast as possible on a pipelined processor like that in lab #4. Note that if you carefully adjust the program you should be able to fill all the delay slots of the program (tricky! Think outside the box).

```
LOOP:
1:  LDUR  X2, [X10, #0]  4:  ↓ RAW x2  ↓ RAW x5
2:  SUB  X4, X2, X3
3:  STUR X4, [X10, #0]
4:  LDUR  X5, [X10, #8]
5:  SUB  X6, X5, X3
6:  STUR X6, [X10, #8]
7:  ADDI X10, X10, #16
8:  CMP  X10, X30
9:  B.NE  LOOP
```

Problem: No instruction to fill 9's delay slot. But, following code does same computation:

- Original
- Alternative

```
6:  STUR X6, [X10, #8]
7:  ADDI X10, X10, #16
```

Alternative constraint graph

Schedule (several possible)

```
3, 4, 2, 5, 3, 6a, 8, 9, 7a
```

Control