# 第一次作业

## 8.2.5

s = 0

i = 0

L1: if i > n goto L2

s = s + i

i = i + 1

goto L1

L2:

上面的三地址代码片段的功能就是求和1+2+…+n

**Long version:**

LD R1, #0 2

ST s, R1 2

ST i, R1 2

L1: LD R1, i 2

LD R2, n 2

SUB R2, R1, R2 1

BGTZ R2, L2 2

LD R2, s 2

ADD R2, R2, R1 1

ST s, R2 2

ADD R1, R1, #1 2

ST i, R1 2

BR L1 2

L2:

指令的存储代价为最后一列数字加和，计算可得指令代价=24

**Short version:**

LD R2, #0 2

LD R1, R2 1

LD R3, n 2

L1: SUB R4, R1, R3 1

BGTZ R4, L2 2

ADD R2, R2, R1 1

ADD R1, R1, #1 2

BR L1 2

L2:

指令的存储代价为最后一列数字加和，计算可得指令代价=13

注：此题答案不固定，只要执行语义和指令代价正确就行

## 8.3.3

(1)

x = a [ i ]

y = b [ j ]

a [ i ] = y

b [ j ] = x

使用栈式分配，生成的代码如下：

LD R1, i

MUL R1, R1, 4

ADD R1, R1, SP

LD R2, a(R1)

ST x(SP), R2 // x = a [ i ]

LD R3, j

MUL R3, R3, 4

ADD R3, R3, SP

LD R4, b(R3)

ST y(SP), R4 // y = b [ j ]

ST a(R1), y(SP) // a [ i ] = y

ST b(R3), x(SP) // b [ j ] = x

# 第二次作业

## 8.4.1

for (i=0; i<n; i++)

for (j=0; j<n; j++)

c[i][j] = 0.0;

for (i=0; i<n; i++)

for (j=0; j<n; j++)

for (k=0; k<n; k++)

c[i][j] = c[i][j] + a[i][k]\*b[k][j];

1)

B1 1) i = 0

B2 2) if i >= n goto(13)

B3 3) j = 0

B4 4) if j >= n goto(11)

B5 5） t1 = n \* i

6) t2 = t1 + j

7) t3 = t2 \* 8

8) c[t3] = 0.0

9) j = j + 1

10) goto(4)

B6 11) i = i + 1

12) goto(2)

B7 13) i = 0

B8 14) if i >= n goto(40)

B9 15) j = 0

B10 16) if j >= n goto(38)

B11 17) k = 0

B12 18) if k >= n goto(36)

B13 19) t4 = n \* i

20) t5 = t4 + j

21) t6 = t5 \* 8

22) t7 = c[t6]

23) t8 = n \* i

24) t9 = t8 + k

25) t10 = t9 \* 8

26) t11 = a[t10]

27) t12 = n \* k

28) t13 = t12 + j

29) t14 = t13 \* 8

30) t15 = b[t14]

31) t16 = t11 \* t15

32) t17 = t7 + t16

33) c[t6] = t17

34) k = k + 1

35) goto(18)

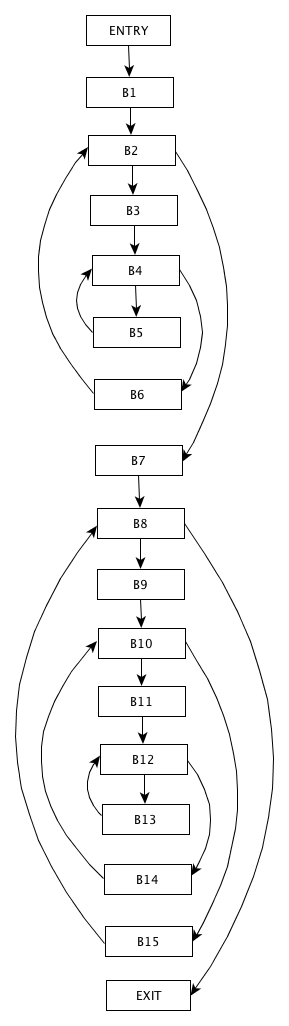
B14 36) j = j + 1

37) goto(16)

B15 38) i = i + 1

39) goto(14)

2)基本块见上图的标号，构造的流图如下：



3)循环有：

{B2,B3,B4,B5,B6}

{B4,B5}

{B8, B9, B10, B11, B12, B13,B14, B15}

{B10, B11, B12, B13,B14}

{B12, B13}

## 8.5.1

d = b + c

e = a + b

b = b \* c

a = e – d

构造的DAG如下：



只有a在基本块的出口活跃：

d=b+c

e=a+b

a=e-d

# 第三次作业

## 8.6.1

(1) x = a + b \* c;

三地址代码：

t1=b\*c

x=a+t1

生成的目标代码：

LD R1, b

LD R2, c

MUL R1,R1,R2

LD R3,a

ADD R1,R1,R3

ST x, R1

(4) a [ i ] = b [ c [ i ] ];

三地址代码：

t1=i\*4

t2=c[t1]

t3=t2\*4

a[t1]=b[t3]

生成的目标代码：

LD R1, i

MUL R1,R1,4

LD R2, c(R1)

MUL R2,R2,4

LD R3, b(R2)

ST a(R1), R3

(6) \*p++ = \*q++

三地址代码：

\*p=\*q

q=q+4

p=p+4

生成的目标代码：

LD R1,q

LD R2,0(R1)

LD R3,p

ST 0(R3), R2

ADD R1,R1,4

ADD R3,R3,4

## 8.8.1

为图8-17的程序构造寄存器冲突图（干涉图）

程序在各个点的活跃变量情况如下图所示：



因此构造的寄存器冲突图如下：



是一个包含6个结点的完全图。