

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 $1+2+\dots+n$

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 $i \geq n$ goto(13)

B3 3) $j = 0$

B4 4) if $j \geq n$ 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 $i \geq n$ goto(40)

B9 15) $j = 0$

B10 16) if $j \geq n$ goto(38)

B11 17) $k = 0$

B12 18) if $k \geq n$ goto(36)

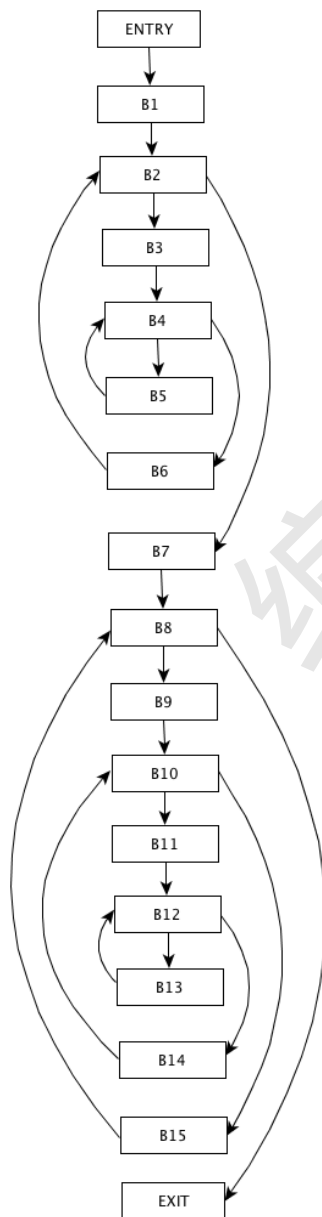
B13 19) $t4 = n * i$
20) $t5 = t4 + j$
21) $t6 = t5 * 8$
22) $t7 = c[t6]$
23) $t8 = n * i$
24) $t9 = t8 + k$
25) $t10 = t9 * 8$
26) $t11 = a[t10]$
27) $t12 = n * k$
28) $t13 = t12 + j$
29) $t14 = t13 * 8$
30) $t15 = b[t14]$
31) $t16 = t11 * t15$
32) $t17 = t7 + t16$

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,B6}

{B4,B5}

{B8, B9, B10, B15}

{B10, B11, B12,,B14}

{B12, B13}

8.5.1

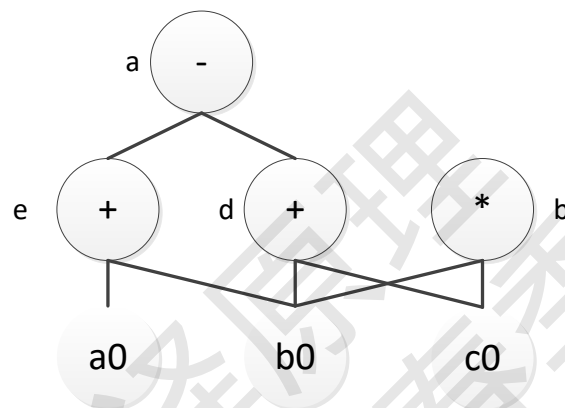
$d = b + c$

$e = a + b$

$b = b * c$

$a = e - d$

构造的 DAG 如下:



只有 a 在基本块的出口活跃:

$d = b + c$

$e = a + b$

$a = e - d$

第四次作业

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

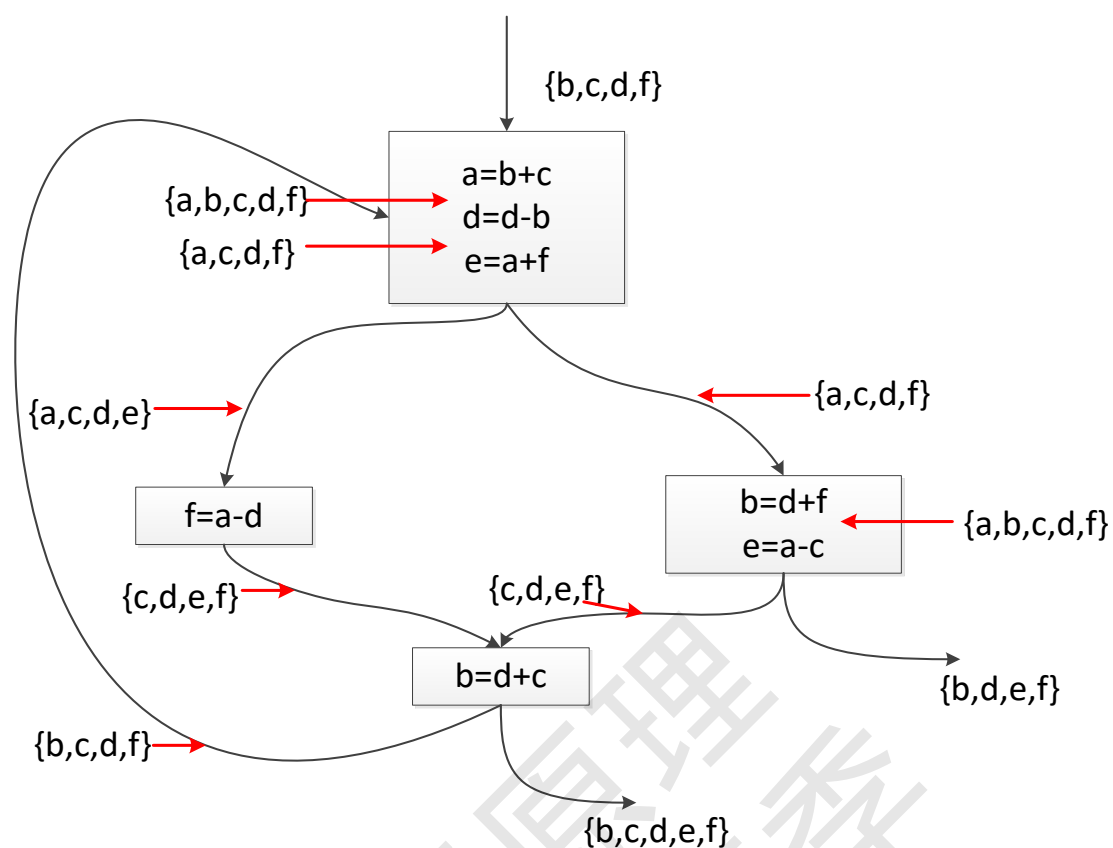
ADD R1,R1,4

ADD R3,R3,4

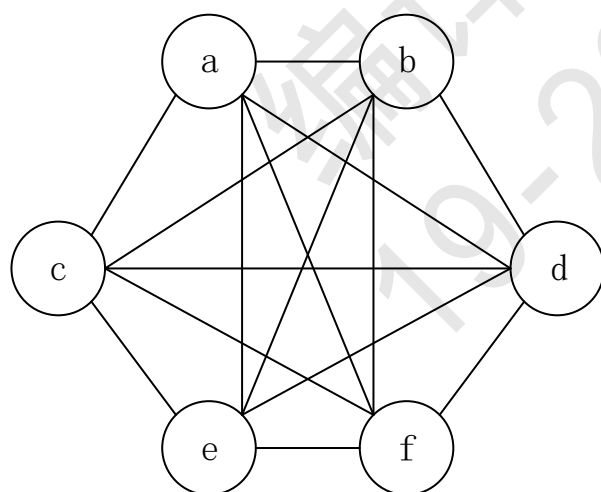
8.8.1

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。

第三次作业

	def-use chain	use-def chain	活跃变量计算		可用表达式	
			def(B) & use(B)	ln(B) & Out(B)	gen(B) & kill(B)	ln(B) & Out(B)
B1 (1)a:=1 (2)b:=2	a, (1), {(3), (4), (6), (8), (9)} b, (2), {(3), (5), (6), (8)}		def(B1)={a, b} use(B1)={}	ln(B1)={e} Out(B1)={a, b, e}	gen(B1)={} kill(B1)={a+b, c-a, b*d, a-d}	ln(B1)= \varnothing Out(B1)= \varnothing
B2 (3)c:=a+b (4)d:=c-a	c, (3), {(4), (9)} d, (4), {(5), (10), (11)}	a, (3), {(1)} b, (3), {(2), (8)} c, (4), {(3)} a, (4), {(1)}	def(B2)={c, d} use(B2)={a, b}	ln(B2)={a, b, e} Out(B2)={a, b, c, d, e}	gen(B2)={a+b, c-a} kill(B2)={b*d, a-d}	ln(B2)= \varnothing Out(B2)={a+b, c-a}
B3 (5)d:=b*d	d, (5), {(10), (11)}	b, (5), {(2), (8)} d, (5), {(4), (6)}	def(B3)={} use(B3)={b, d}	ln(B3)={a, b, c, d, e} Out(B3)={a, b, c, d, e}	gen(B3)={ \varnothing } kill(B3)={b*d, a-d}	ln(B3)={a+b, c-a} Out(B3)={a+b, c-a}
B4 (6)d:=a+b (7)e:=e+1	d, (6), {(5)} e, (7), {(7)}	a, (6), {(1)} b, (6), {(2), (8)} e, (7), {(7), (9)}	def(B4)={d} use(B4)={a, b, e}	ln(B4)={a, b, c, e} Out(B4)={a, b, c, d, e}	gen(B4)={a+b} kill(B4)={a-d, b*d, e+1}	ln(B4)={a+b, c-a} Out(B4)={a+b, c-a}
B5 (8)b:=a+b (9)e:=c-a	b, (8), {(3), (5), (6), (8), (10)} e, (9), {(7)}	a, (8), {(1)} b, (8), {(2), (8)} c, (9), {(3)} a, (9), {(1)}	def(B5)={e} use(B5)={a, b, e}	ln(B5)={a, b, c, d} Out(B5)={a, b, d, e}	gen(B5)={c-a} kill(B5)={a+b, b*d, e+1}	ln(B5)={a+b, c-a} Out(B5)={c-a}
B6 (10)a:=b*d (11)b:=a-d	e, (10), {(11)} b, (11), {}	b, (10), {(8)} d, (10), {(4), (5)} a, (11), {(10)} d, (11), {(4), (5)}	def(B6)={a} use(B6)={b, d}	ln(B6)={b, d} Out(B6)={ \varnothing }	gen(B6)={a-d} kill(B6)={a+b, c-a, b*d}	ln(B6)={c-a} Out(B6)={a-d}

说明: a, (1), {(3),(4),(6),(8),(9)}表示 a 在 (1) 的 def-use 链是{(3),(4),(6),(8),(9)}, a, (3), {(1)}表示 a 在 (3) 的 use-def 链是{(1)}

可用表达式计算中, $U=\{a+b, c-a, b*d, e+1, a-d\}$