

1. 对以下中间代码序列 C:

T1:=B-C

T2:=A*T1

T3:=D+1

T4:=E-F

T5:=T3*T4

W:=T2/T5

假设可用寄存器为 R0 和 R1，W 是基本块出口的活跃变量，用简单代码生成算法生成其目标代码，同时列出代码生成过程中的寄存器描述和地址描述。

解答：

生成的目标代码、寄存器描述和地址描述如下：

| 中间代码 | 目标代码 | 寄存器描述 | 地址描述 |
|-----------|-----------------------------------|--------------------|------------------------------|
| T1:=B-C | LD R0,B SUB R0,C | R0 含 T1 | T1 在 R0 |
| T2:=A*T1 | LD R1,A MUL R1,R0 | R0 含 T1 R1 含 T2 | T1 在 R0 T2 在 R1 |
| T3:=D+1 | LD R0,D ADD R0,#1 | R1 含 T2 R0 含 T3 | T2 在 R1 T3 在 R0 |
| T4:=E-F | ST R1,T2 LD R1,E SUB R1,F | R0 含 T3 R1 含 T4 | T3 在 R0 T4 在 R1 T2 在内存 |
| T5:=T3*T4 | MUL R0,R1 | R1 含 T4 R0 含 T5 | T4 在 R1 T2 在内存 T5 在 R0 |
| W:=T2/T5 | LD R1,T2 DIV R1,R0 ST R1, W | R0 含 T5 R1 含 W | T5 在 R0 W 在内存和 R1 |

2. 对以下中间代码序列：

T1:=A+B

T2:=T1-C

$T3 := D + E$

$T3 := T2 * T3$

$T4 := T1 + T3$

$T5 := T3 - E$

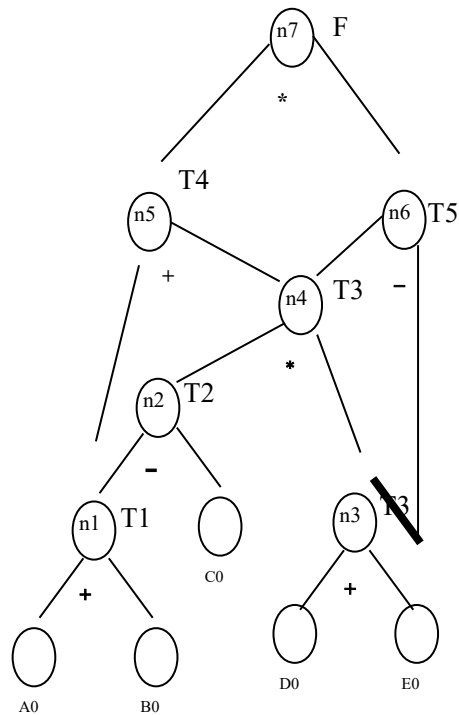
$F := T4 * T5$

(1) 应用 DAG 结点排序算法重新排序；

(2) 假设可用寄存器为 R0, F 是基本块出口处的活跃变量，应用简单代码生成算法分别生成排序前后的中间代码序列的目标代码，并比较其优劣。

解答：

(1) 设原中间代码序列为 G，则 G 的 DAG 图为：



由上可知 DAG 图有 7 个内部结点，根据结点重新排序算法，将内部结点进行排序，为：

$n3, n1, n2, n4, n6, n5, n7$

按照该顺序将中间代码进行改写得到 G'：

$T3 := D + E$

$T1 := A + B$

$T2 := T1 - C$

$T3 := T2 * T3$

T5:=T3-E

T4:=T1+T3

F:=T4*T5

(2)G 和 G'的目标代码分别为：

- | | |
|----------------|----------------|
| (1) LD R0,A | (1) LD R0,D |
| (2) ADD R0,B | (2) ADD R0,E |
| (3) ST R0,T1 | (3) ST R0,T3 |
| (4) SUB R0,C | (4) LD R0,A |
| (5) ST R0,T2 | (5) ADD R0,B |
| (6) LD R0,D | (6) ST R0,T1 |
| (7) ADD R0,E | (7) ADD R0,C |
| (8) ST R0,T3 | (8) MUL R0,T3 |
| (9) LD R0,T2 | (9) ST R0,T3 |
| (10) MUL R0,T3 | (10) SUB R0,E |
| (11) ST R0,T3 | (11) ST R0 T5 |
| (12) LD R0,T1 | (12) LD R0,T1 |
| (13) ADD R0,T3 | (13) ADD R0,T3 |
| (14) ST R0,T4 | (14) MUL R0,T5 |
| (15) LD R0,T3 | (15) ST R0,F |
| (16) SUB R0,E | |
| (17) ST R0,T5 | |
| (18) LD R0,T4 | |
| (19) MUL R0,T5 | |
| (20) ST R0,F | |

G'的目标代码比 G 的目标代码短，少 5 条指令，说明结点重排后得到代码更优化，更高效。