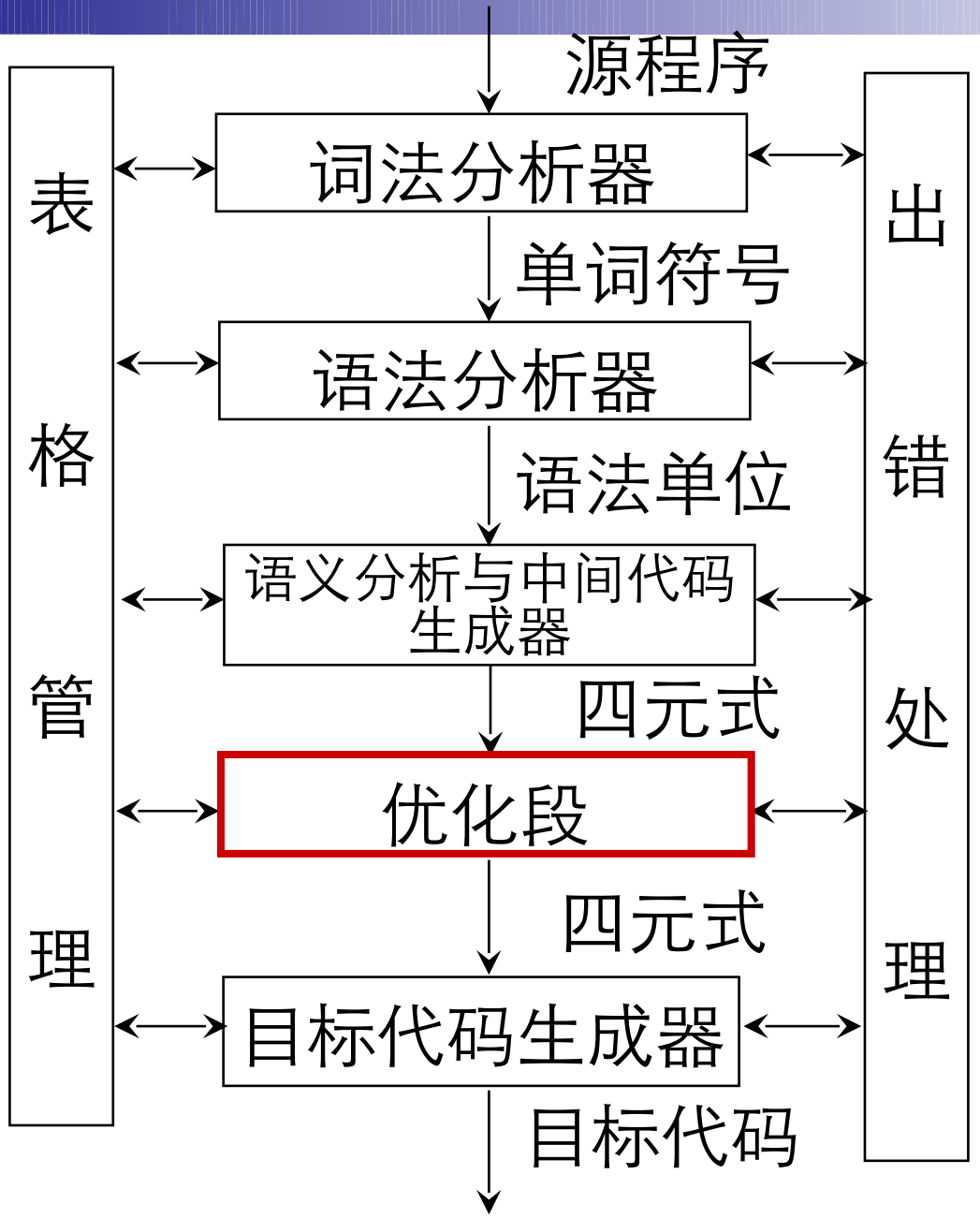
编译原理

第十章 优化

编译程序总框


源程序
词法分析器
单词符号
语法分析器
语法单位
语义分析与中间代码生成器
四元式
优化段
四元式
目标代码生成器
目标代码
表格管理
出错处理

第十章 优化

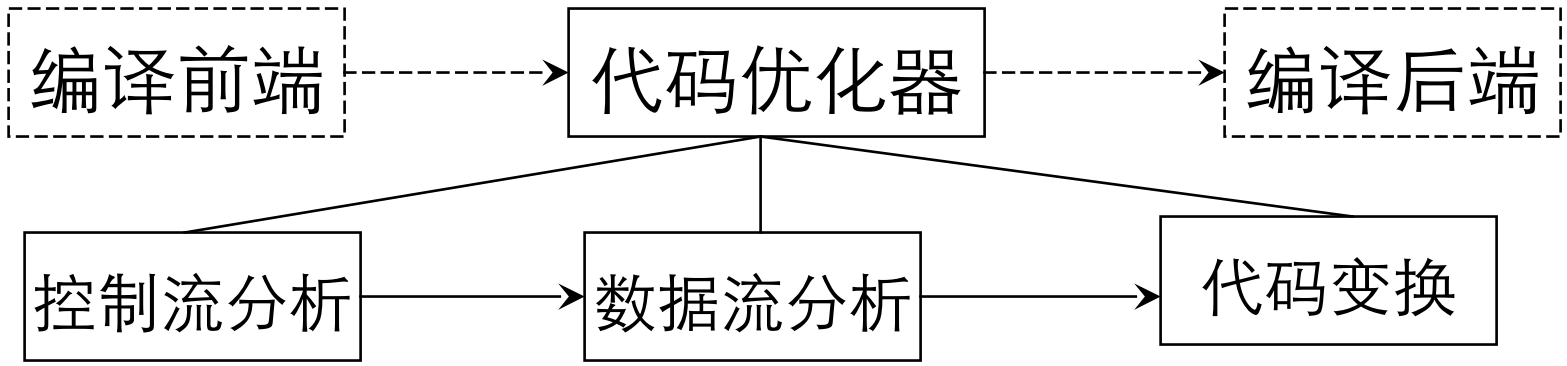
- 优化概述
- 局部优化
- 循环优化

第十章 优化

- 优化概述
- 局部优化
- 循环优化

10.1 概述

- 优化：对程序进行各种等价变换，使得从变换后的程序出发，能生成更有效的目标代码。
  - 等价：不改变程序的运行结果
  - 有效：目标代码运行时间短，占用存储空间小



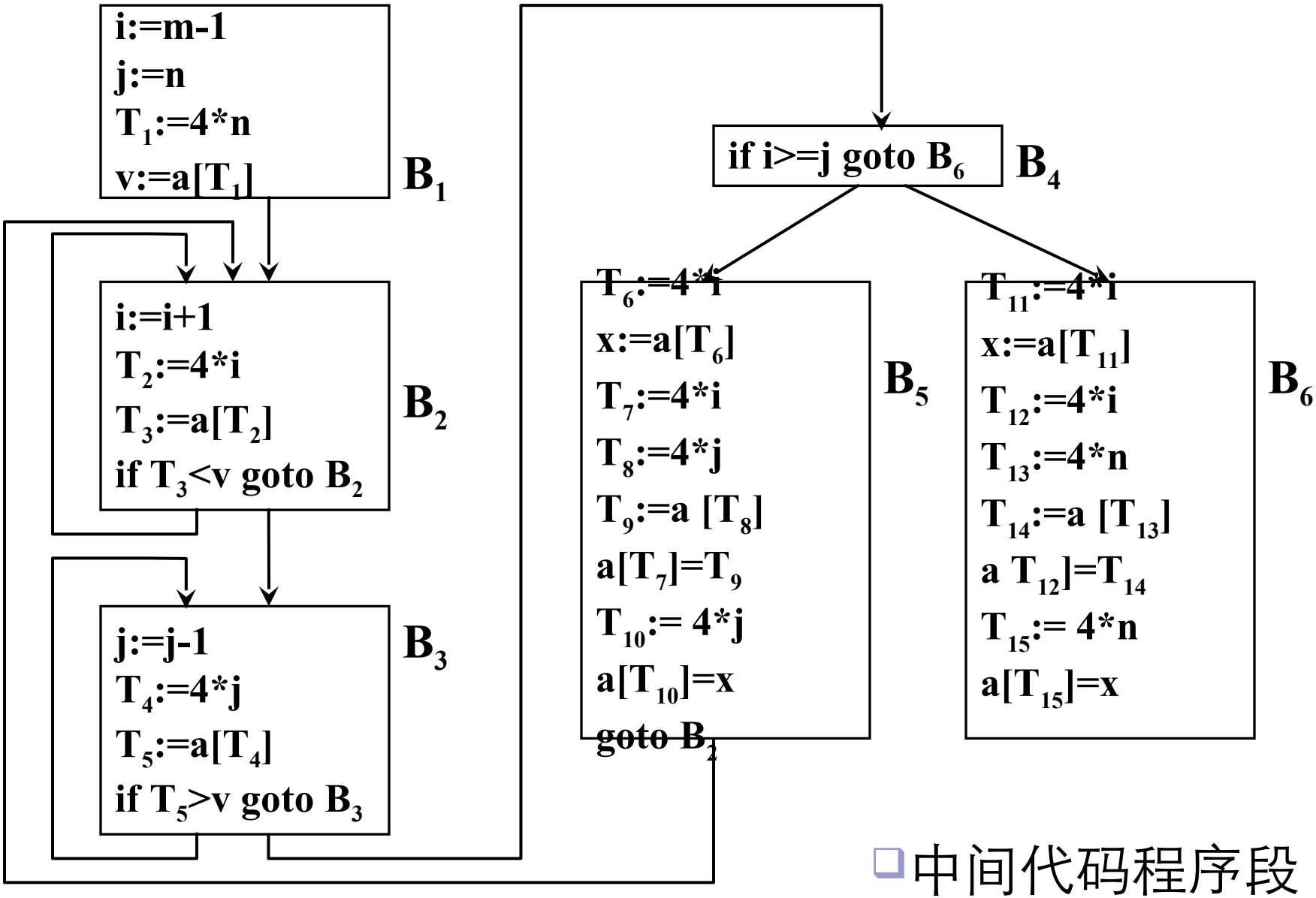
10.1 概述

- 优化的目的
  - 产生更高效的代码
- 优化必须遵循一定的原则
  - 等价原则
    - 经过优化后不应改变程序运行的结果
  - 有效原则
    - 使优化后所产生的目标代码运行时间较短，占用的存储空间较小
  - 合算原则
    - 应尽可能以较低的代价取得较好的优化效果

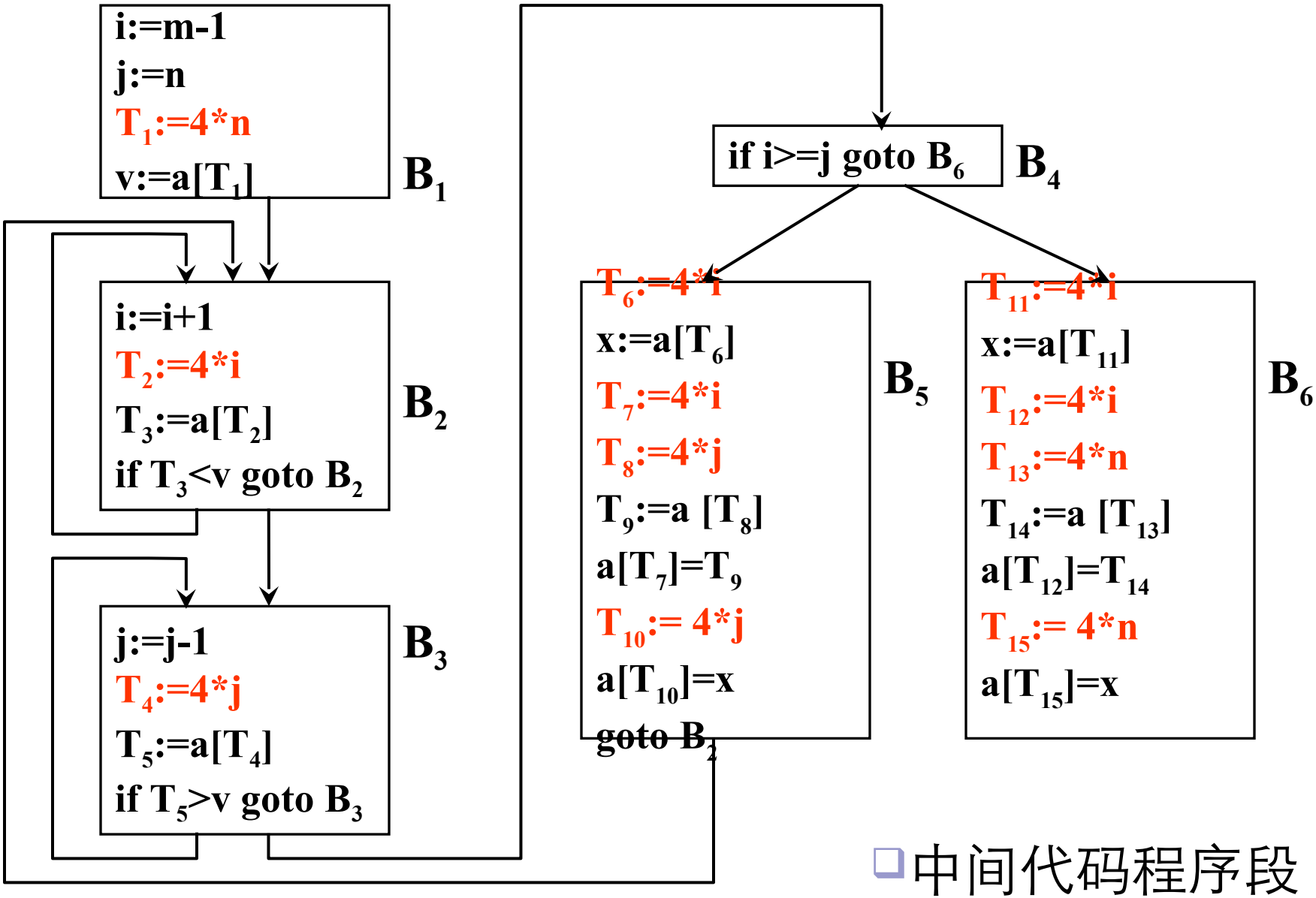
10.1 概述

- 优化的三个不同级别
  - 局部优化
  - 循环优化
  - 全局优化
- 优化的种类
  - 删除多余运算（或称删除公用子表达式）
  - 合并已知量
  - 复写传播
  - 删除无用赋值
  - 代码外提
  - 强度消弱
  - 变换循环控制条件

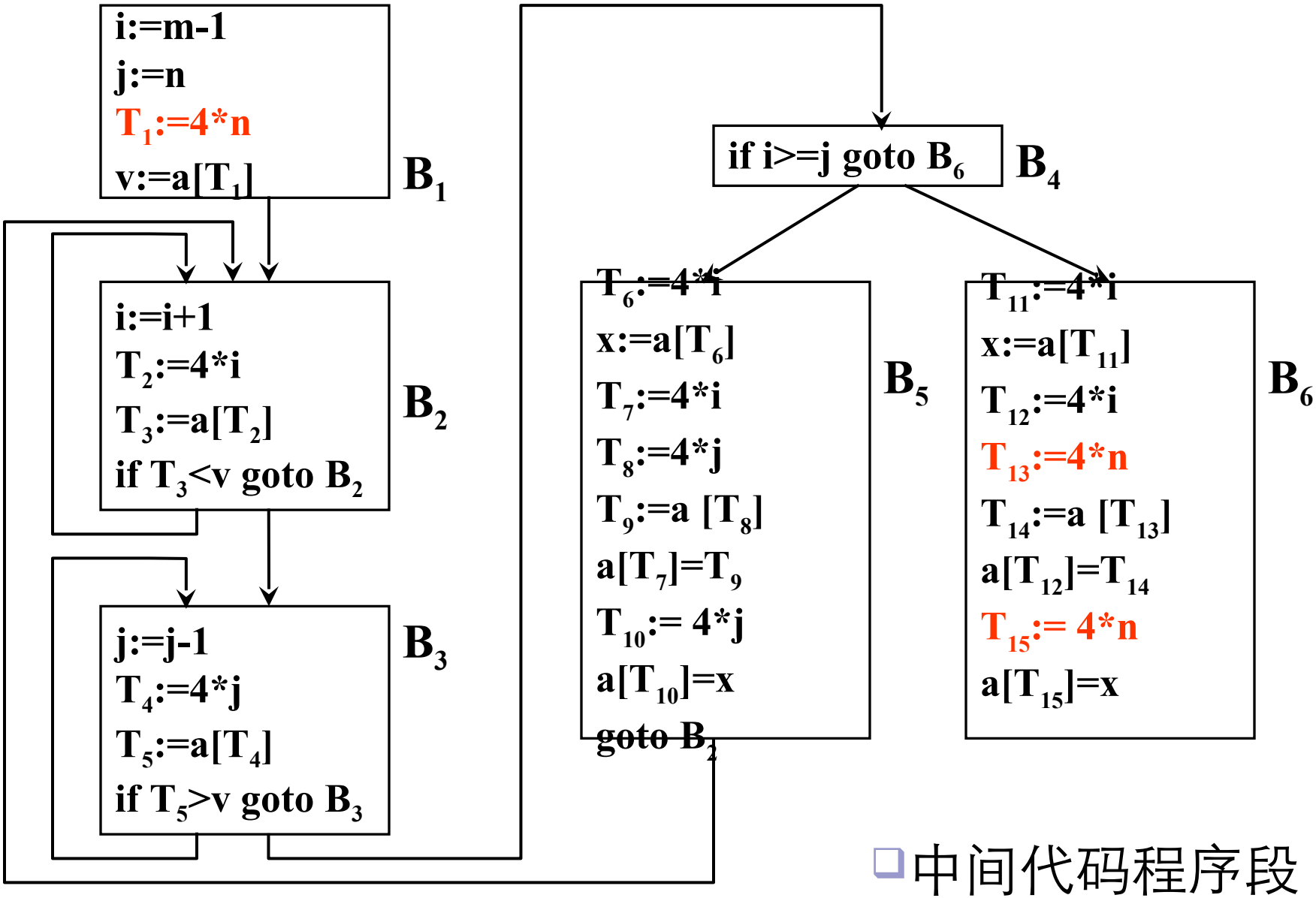
```
void quicksort (m, n);
int m, n;
   {
        int i, j;
        int v, x;
        if (n<=m) return;
        /* fragment begins here*/
        i=m-1; j=n; v=a [n];
        while (1) {
                do  i=i+1;  while (a [i]<v);
                do  j=j-1;  while (a [j]>v);
                if (i>=j) break;
                x=a [i]; a[i]=a [j]; a[j]=x;
         }
       x=a[i]; a[i]=a [n]; a [n]=x;
        /*fragment ends here*/
       quicksort (m, j); quicksort (i+1, n);
}
```


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:=4*i
x:=a[T6]
T7:=4*i
T8:=4*j
T9:=a [T8]
a[T7]=T9
T10:= 4*j
a[T10]=x
goto B2
B5
T11:=4*i
x:=a[T11]
T12:=4*i
T13:=4*n
T14:=a [T13]
a T12]=T14
T15:= 4*n
a[T15]=x
B6

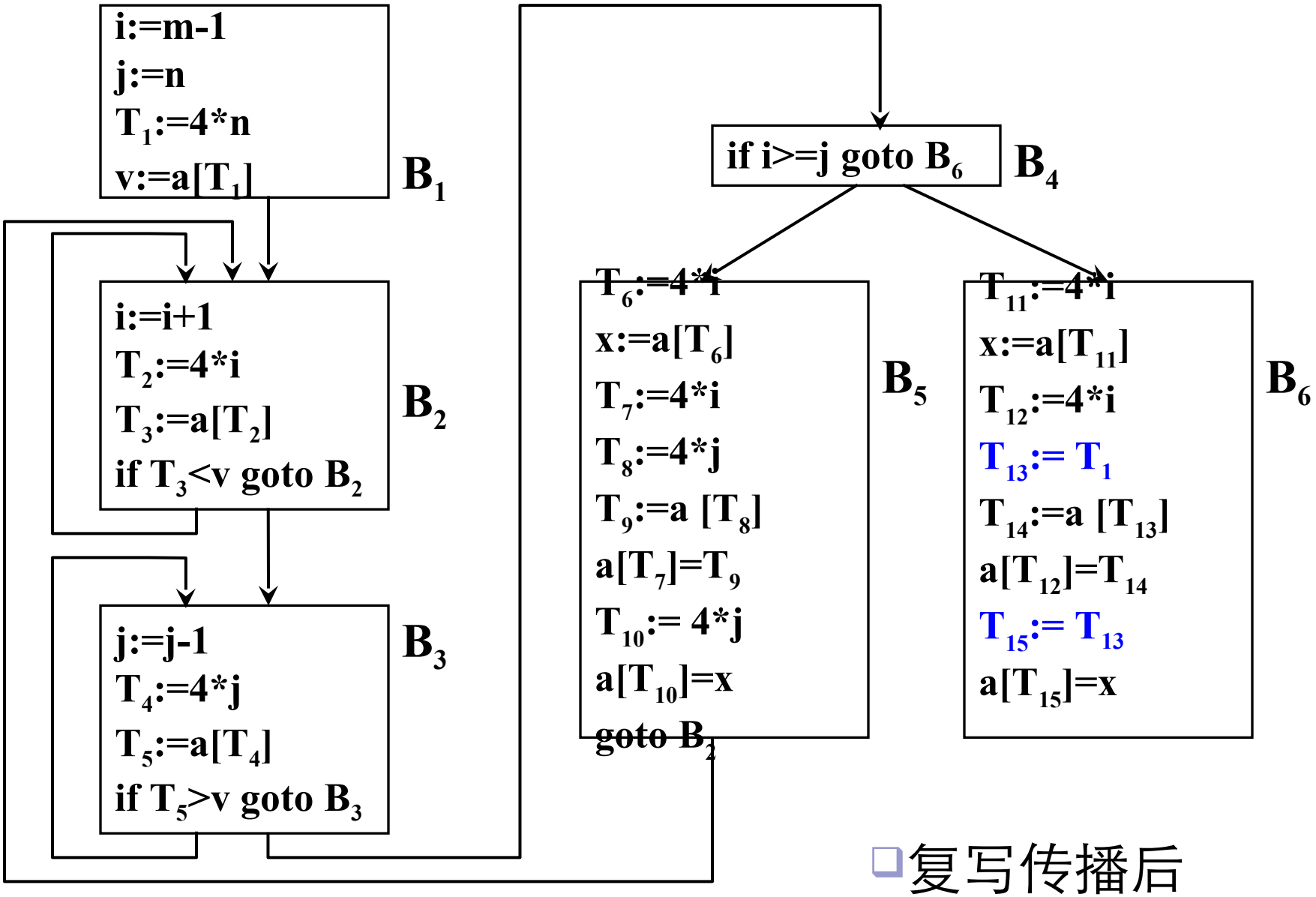
- 中间代码程序段


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:=4*i
x:=a[T6]
T7:=4*i
T8:=4*j
T9:=a [T8]
a[T7]=T9
T10:= 4*j
a[T10]=x
goto B2
B5
T11:=4*i
x:=a[T11]
T12:=4*i
T13:=4*n
T14:=a [T13]
a[T12]=T14
T15:= 4*n
a[T15]=x
B6

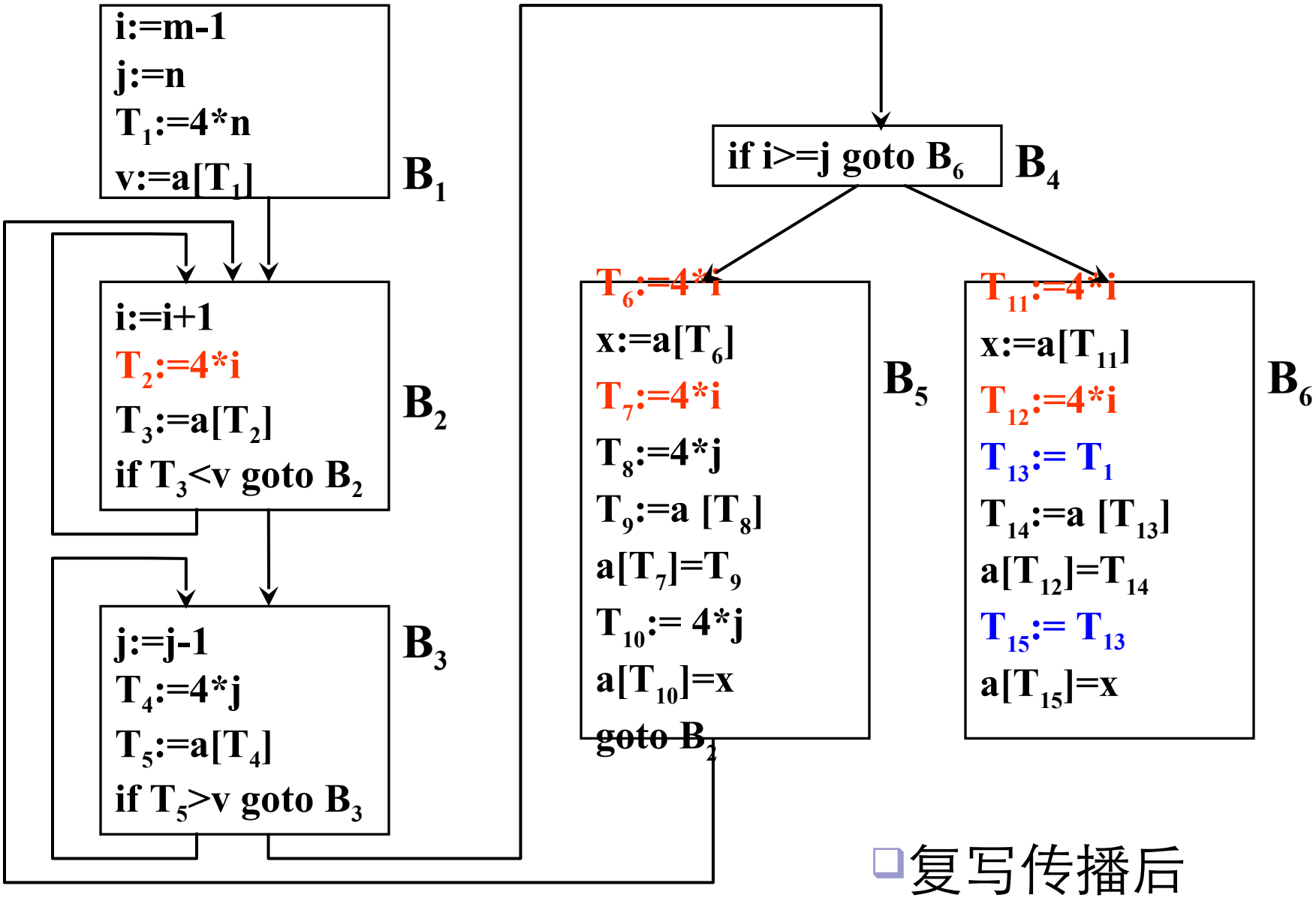
- 中间代码程序段


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:=4*i
x:=a[T6]
T7:=4*i
T8:=4*j
T9:=a [T8]
a[T7]=T9
T10:= 4*j
a[T10]=x
goto B2
B5
T11:=4*i
x:=a[T11]
T12:=4*i
T13:=4*n
T14:=a [T13]
a[T12]=T14
T15:= 4*n
a[T15]=x
B6

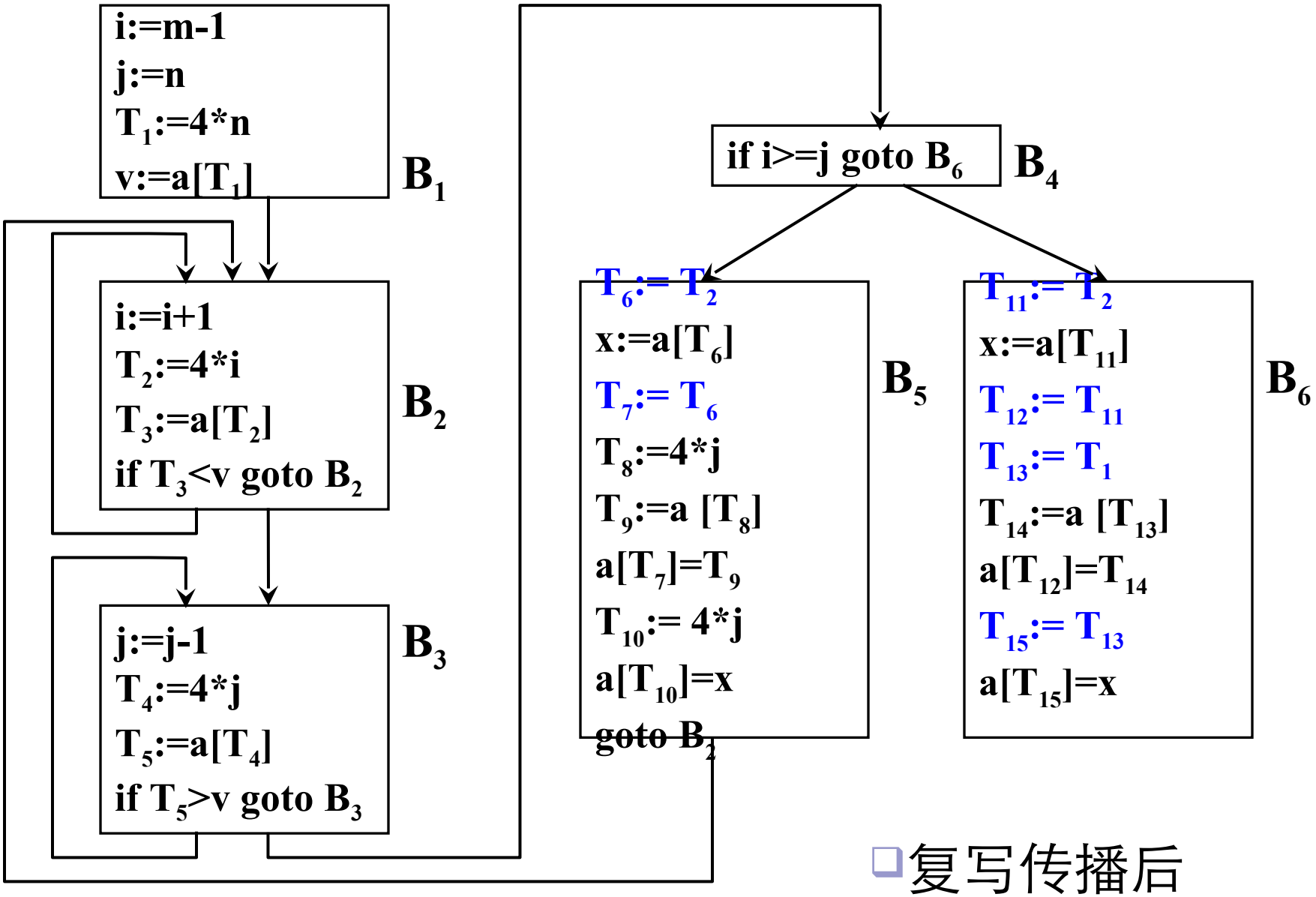
- 中间代码程序段


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:=4*i
x:=a[T6]
T7:=4*i
T8:=4*j
T9:=a [T8]
a[T7]=T9
T10:= 4*j
a[T10]=x
goto B2
B5
T11:=4*i
x:=a[T11]
T12:=4*i
T13:= T1
T14:=a [T13]
a[T12]=T14
T15:= T13
a[T15]=x
B6

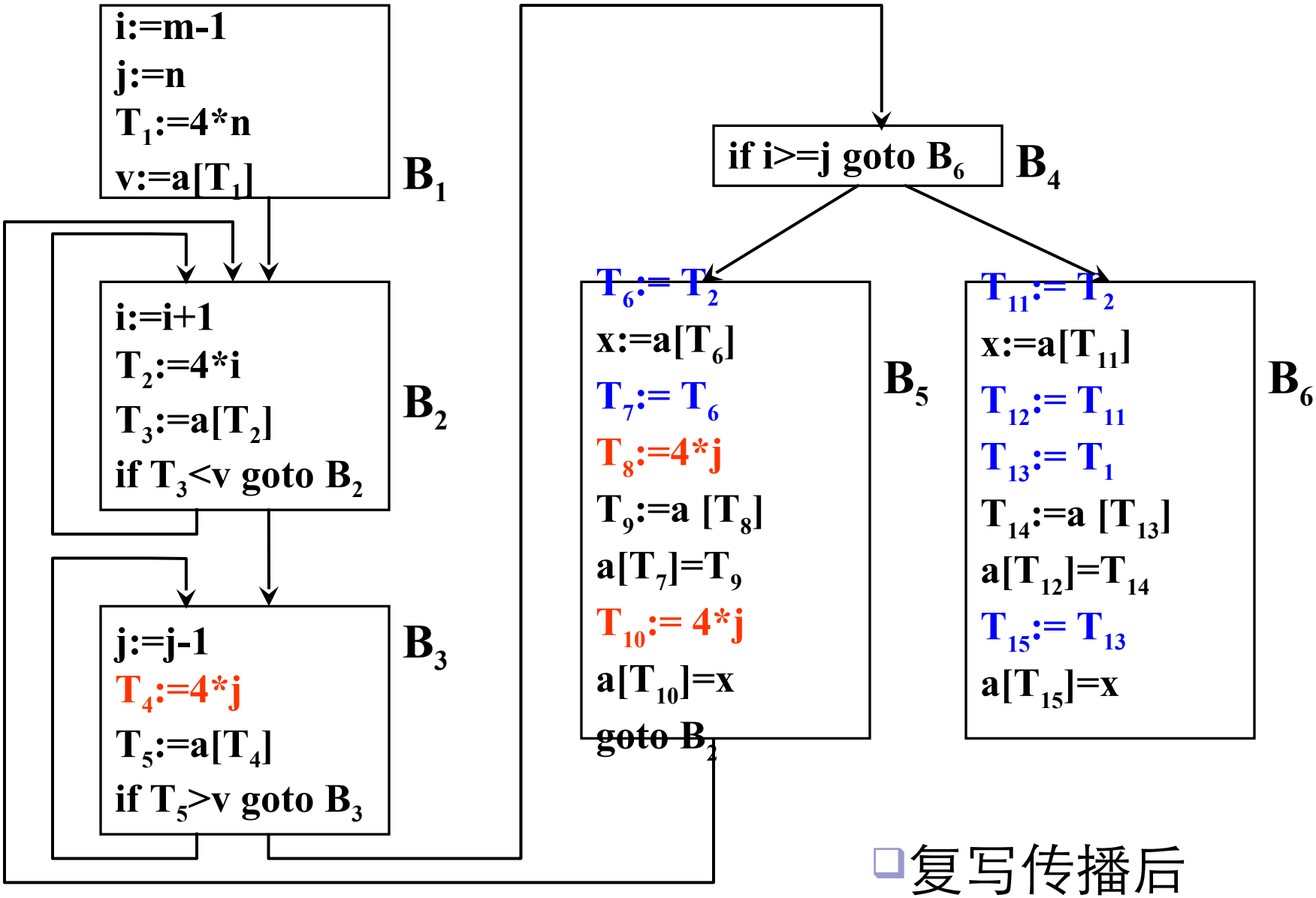
- 复写传播后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:=4*i
x:=a[T6]
T7:=4*i
T8:=4*j
T9:=a [T8]
a[T7]=T9
T10:= 4*j
a[T10]=x
goto B2
B5
T11:=4*i
x:=a[T11]
T12:=4*i
T13:= T1
T14:=a [T13]
a[T12]=T14
T15:= T13
a[T15]=x
B6

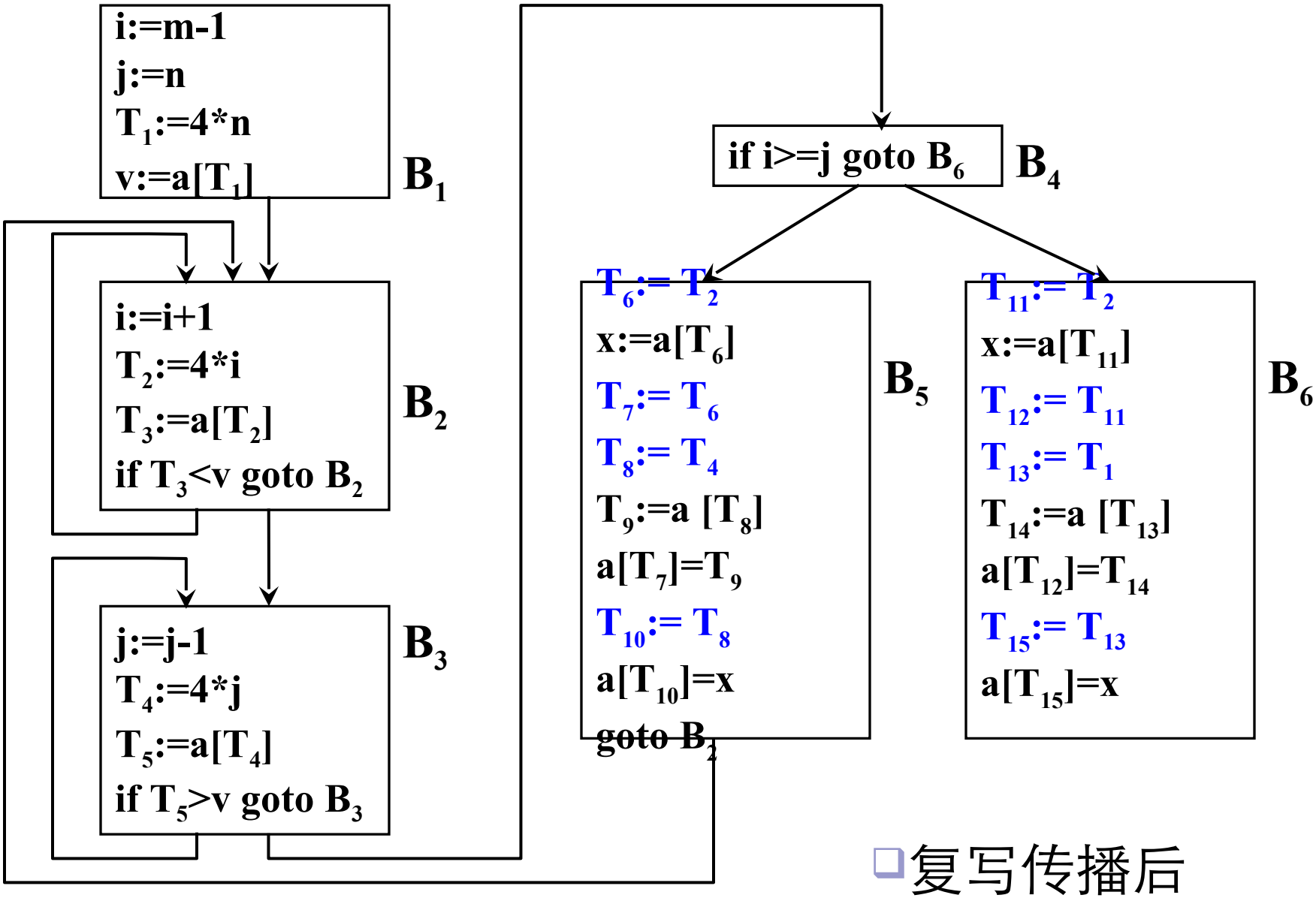
- 复写传播后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:= T2
x:=a[T6]
T7:= T6
T8:=4*j
T9:=a [T8]
a[T7]=T9
T10:= 4*j
a[T10]=x
goto B2
B5
T11:= T2
x:=a[T11]
T12:= T11
T13:= T1
T14:=a [T13]
a[T12]=T14
T15:= T13
a[T15]=x
B6

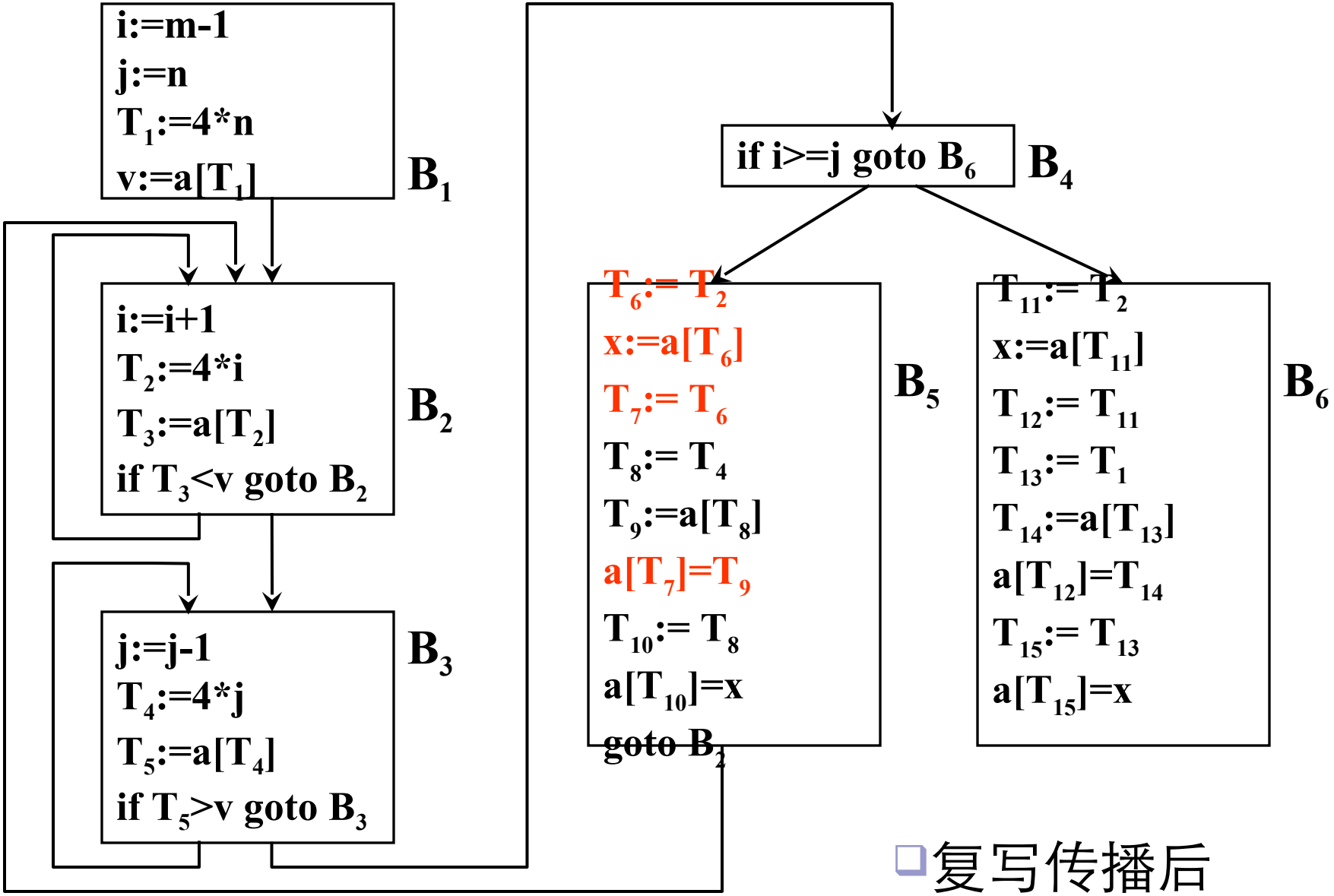
- 复写传播后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:= T2
x:=a[T6]
T7:= T6
T8:=4*j
T9:=a [T8]
a[T7]=T9
T10:= 4*j
a[T10]=x
goto B2
B5
T11:= T2
x:=a[T11]
T12:= T11
T13:= T1
T14:=a [T13]
a[T12]=T14
T15:= T13
a[T15]=x
B6

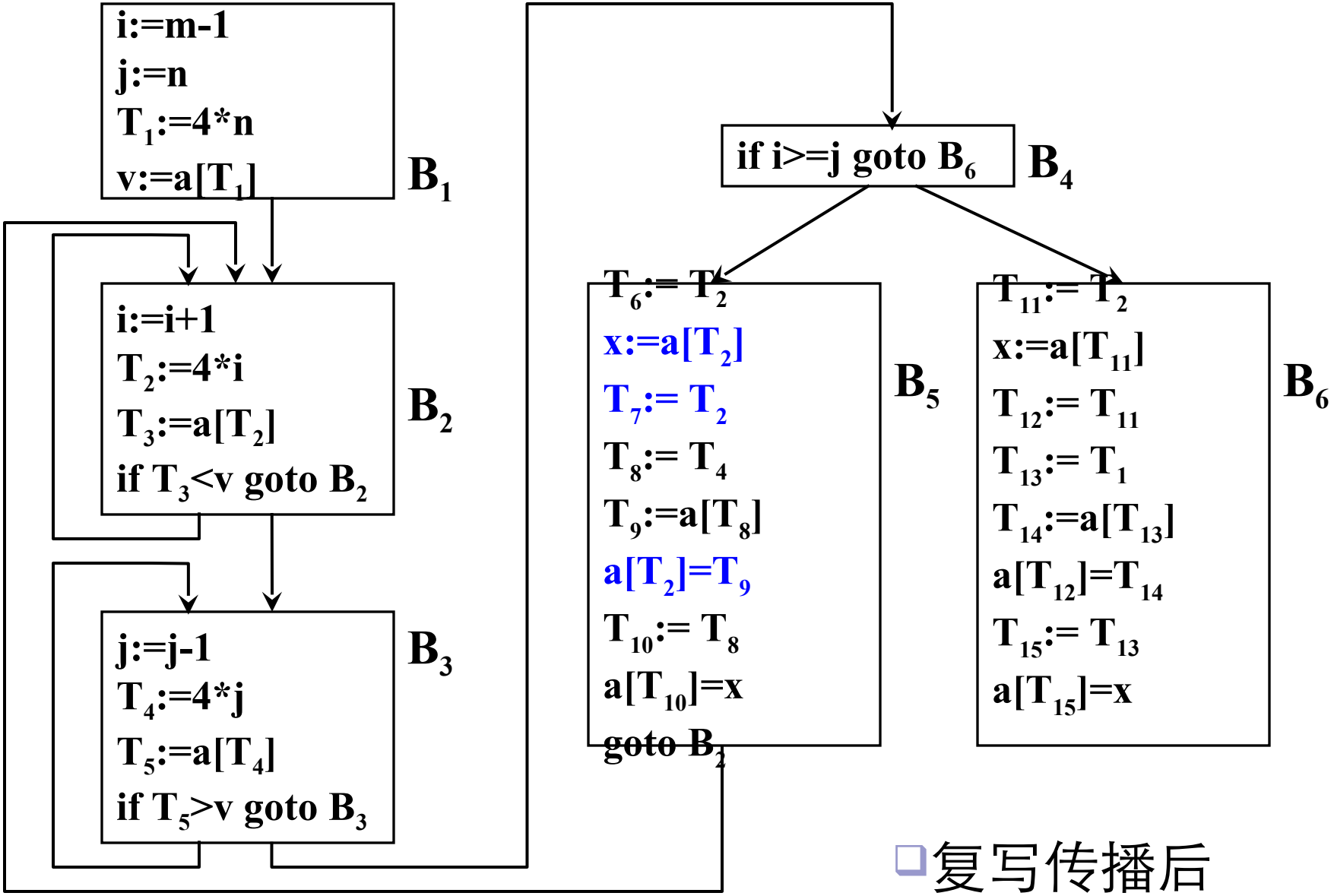
- 复写传播后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:= T2
x:=a[T6]
T7:= T6
T8:= T4
T9:=a [T8]
a[T7]=T9
T10:= T8
a[T10]=x
goto B2
B5
T11:= T2
x:=a[T11]
T12:= T11
T13:= T1
T14:=a [T13]
a[T12]=T14
T15:= T13
a[T15]=x
B6

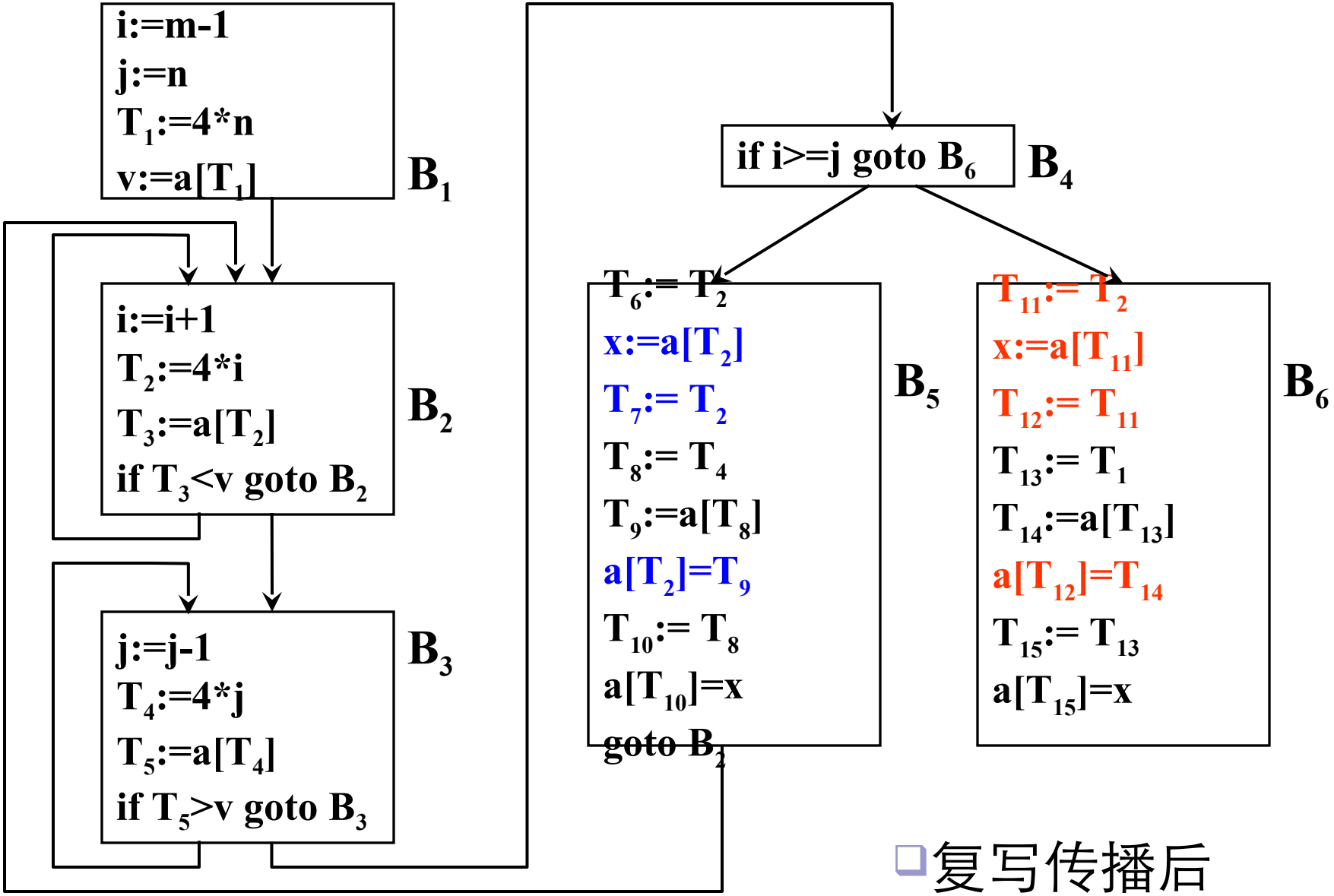
- 复写传播后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:= T2
x:=a[T6]
T7:= T6
T8:= T4
T9:=a[T8]
a[T7]=T9
T10:= T8
a[T10]=x
goto B2
B5
T11:= T2
x:=a[T11]
T12:= T11
T13:= T1
T14:=a[T13]
a[T12]=T14
T15:= T13
a[T15]=x
B6

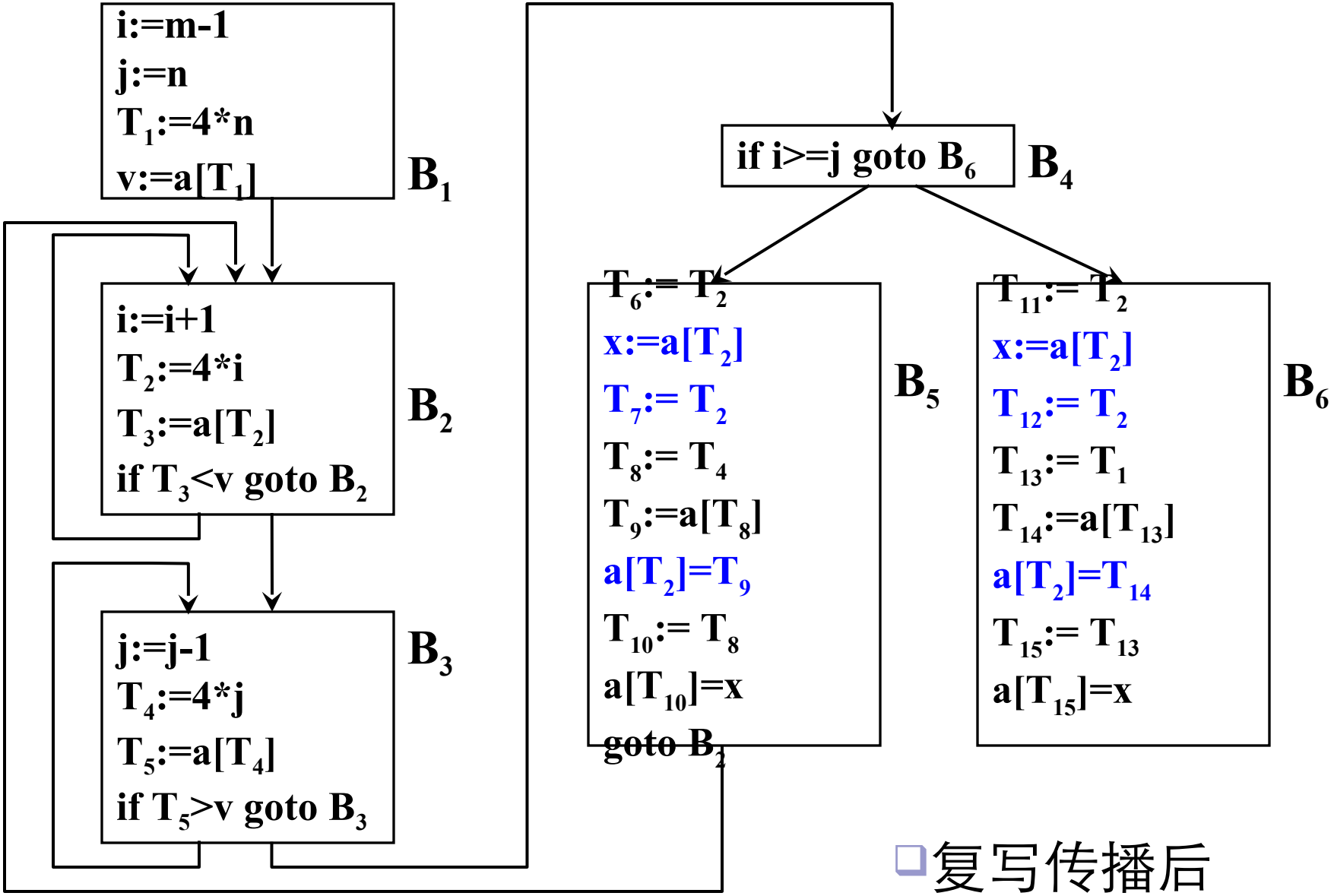
- 复写传播后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:= T2
x:=a[T2]
T7:= T2
T8:= T4
T9:=a[T8]
a[T2]=T9
T10:= T8
a[T10]=x
goto B2
B5
T11:= T2
x:=a[T11]
T12:= T11
T13:= T1
T14:=a[T13]
a[T12]=T14
T15:= T13
a[T15]=x
B6

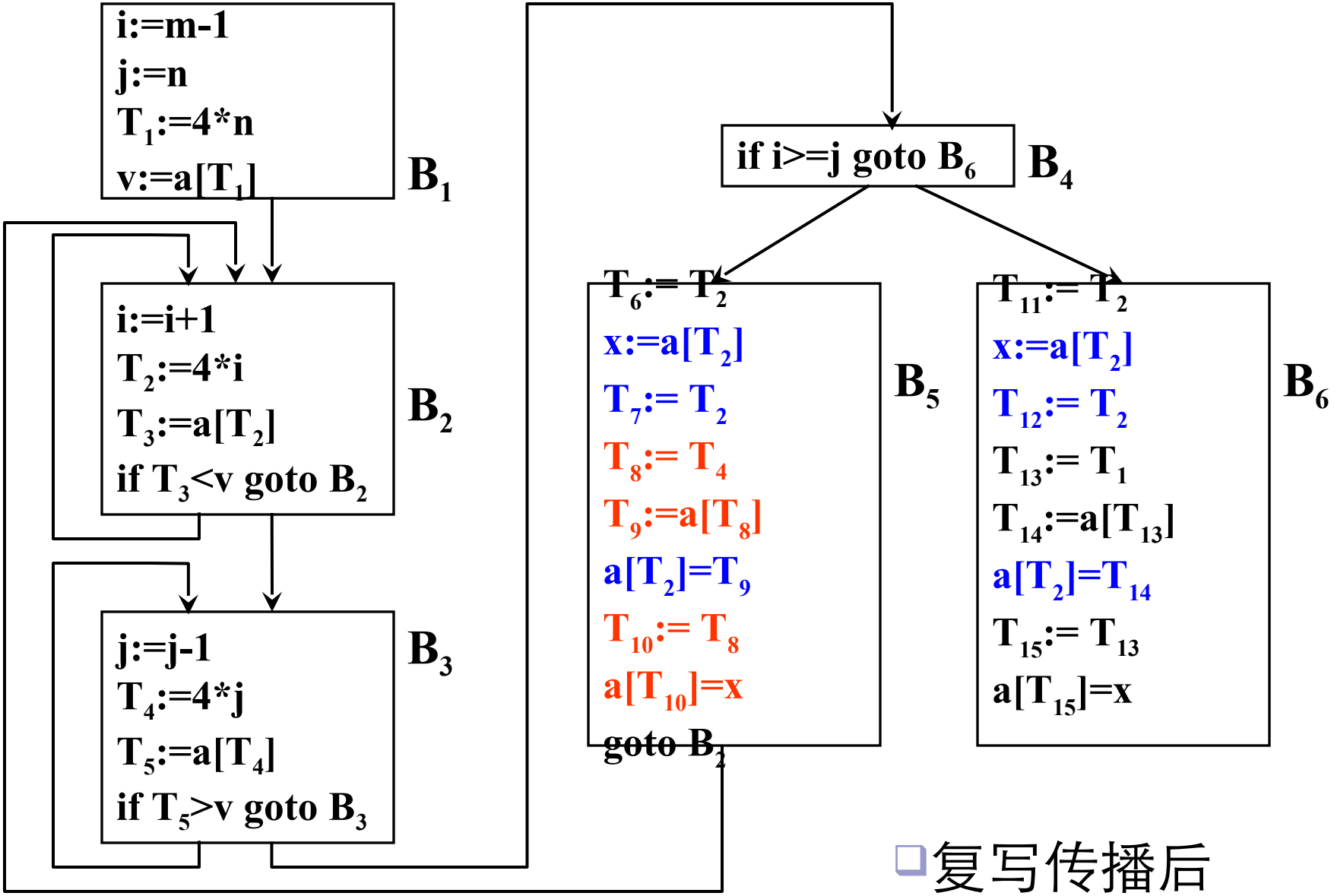
- 复写传播后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:= T2
x:=a[T2]
T7:= T2
T8:= T4
T9:=a[T8]
a[T2]=T9
T10:= T8
a[T10]=x
goto B2
B5
T11:= T2
x:=a[T11]
T12:= T11
T13:= T1
T14:=a[T13]
a[T12]=T14
T15:= T13
a[T15]=x
B6

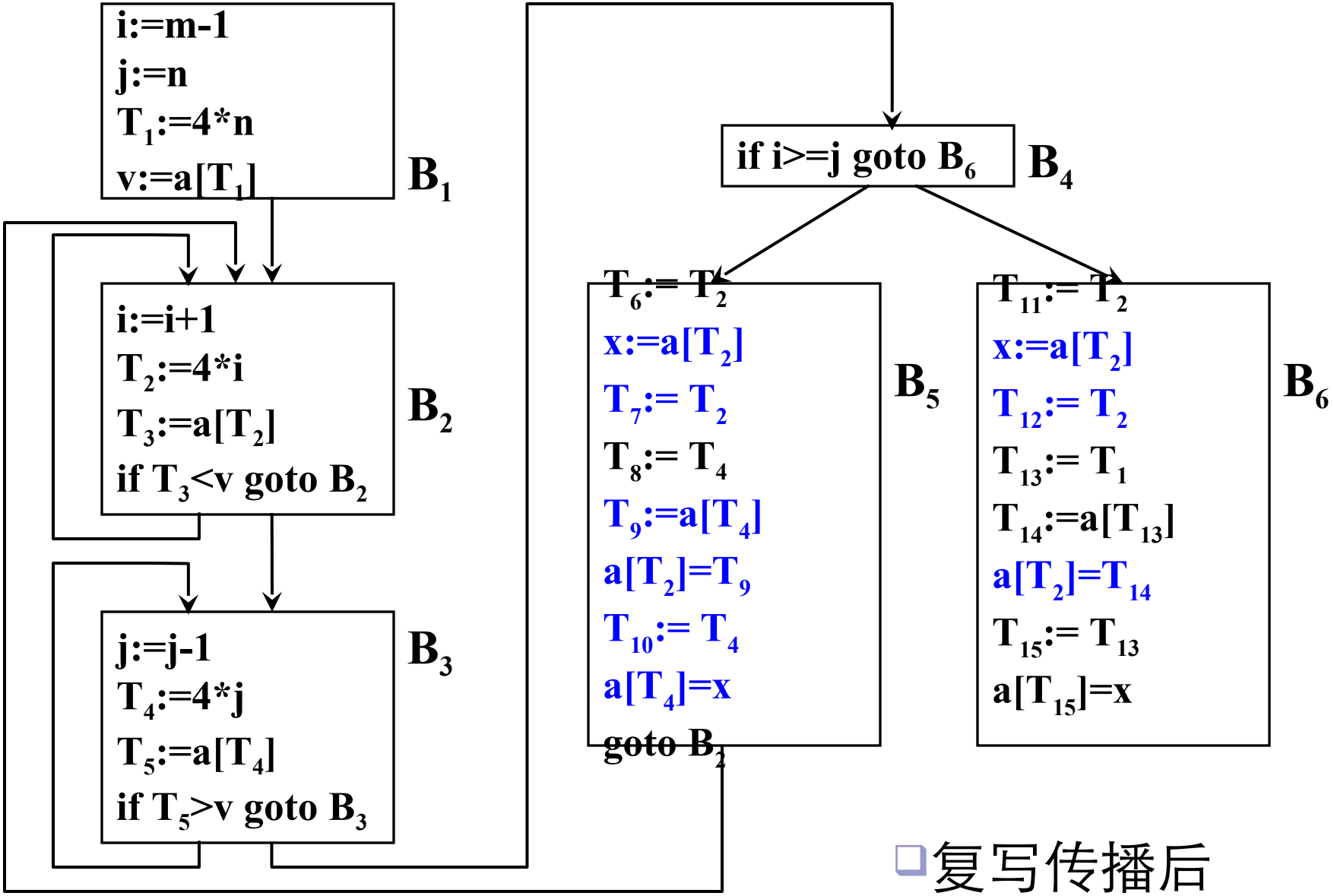
- 复写传播后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:= T2
x:=a[T2]
T7:= T2
T8:= T4
T9:=a[T8]
a[T2]=T9
T10:= T8
a[T10]=x
goto B2
B5
T11:= T2
x:=a[T2]
T12:= T2
T13:= T1
T14:=a[T13]
a[T2]=T14
T15:= T13
a[T15]=x
B6

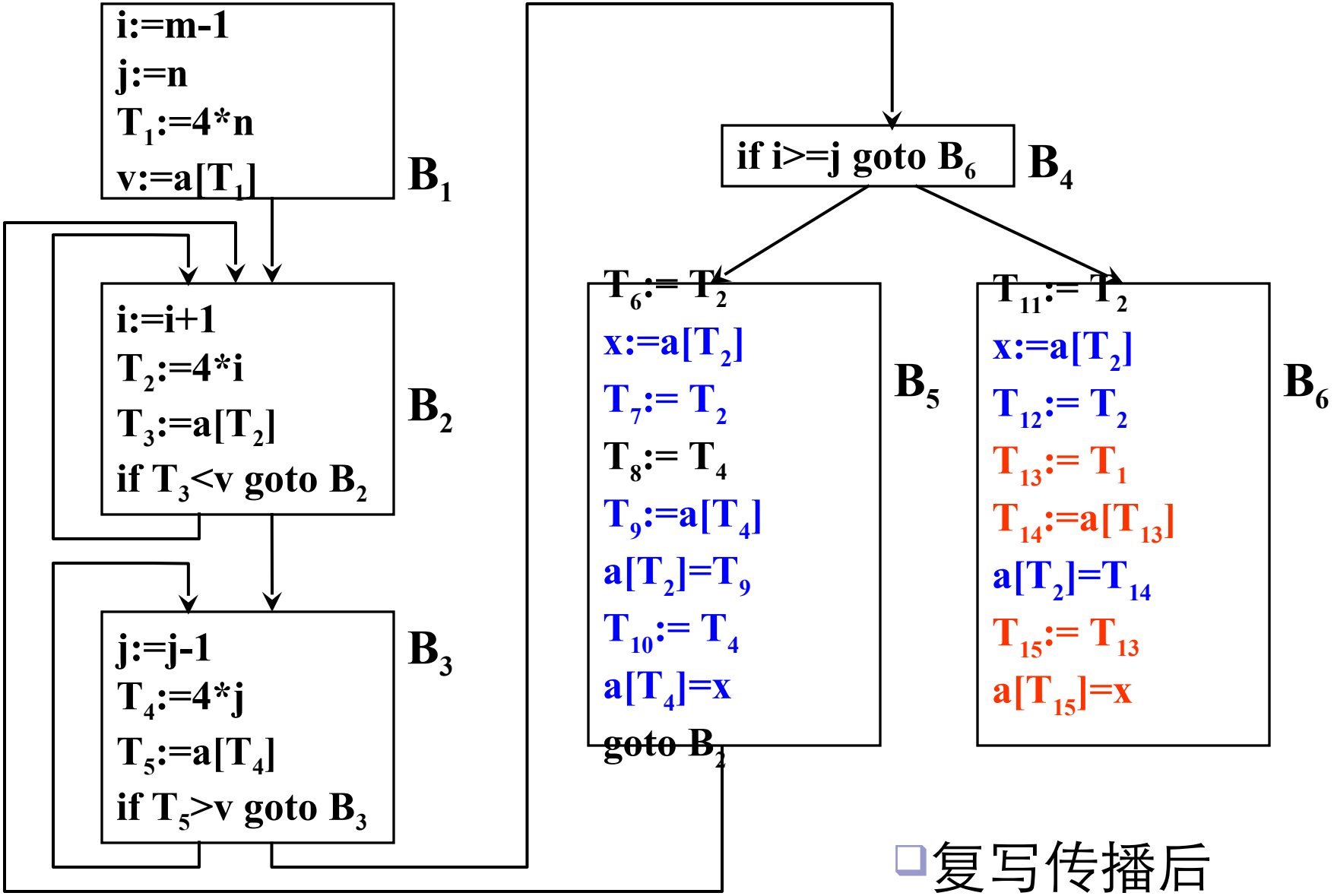
- 复写传播后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:= T2
x:=a[T2]
T7:= T2
T8:= T4
T9:=a[T8]
a[T2]=T9
T10:= T8
a[T10]=x
goto B2
B5
T11:= T2
x:=a[T2]
T12:= T2
T13:= T1
T14:=a[T13]
a[T2]=T14
T15:= T13
a[T15]=x
B6

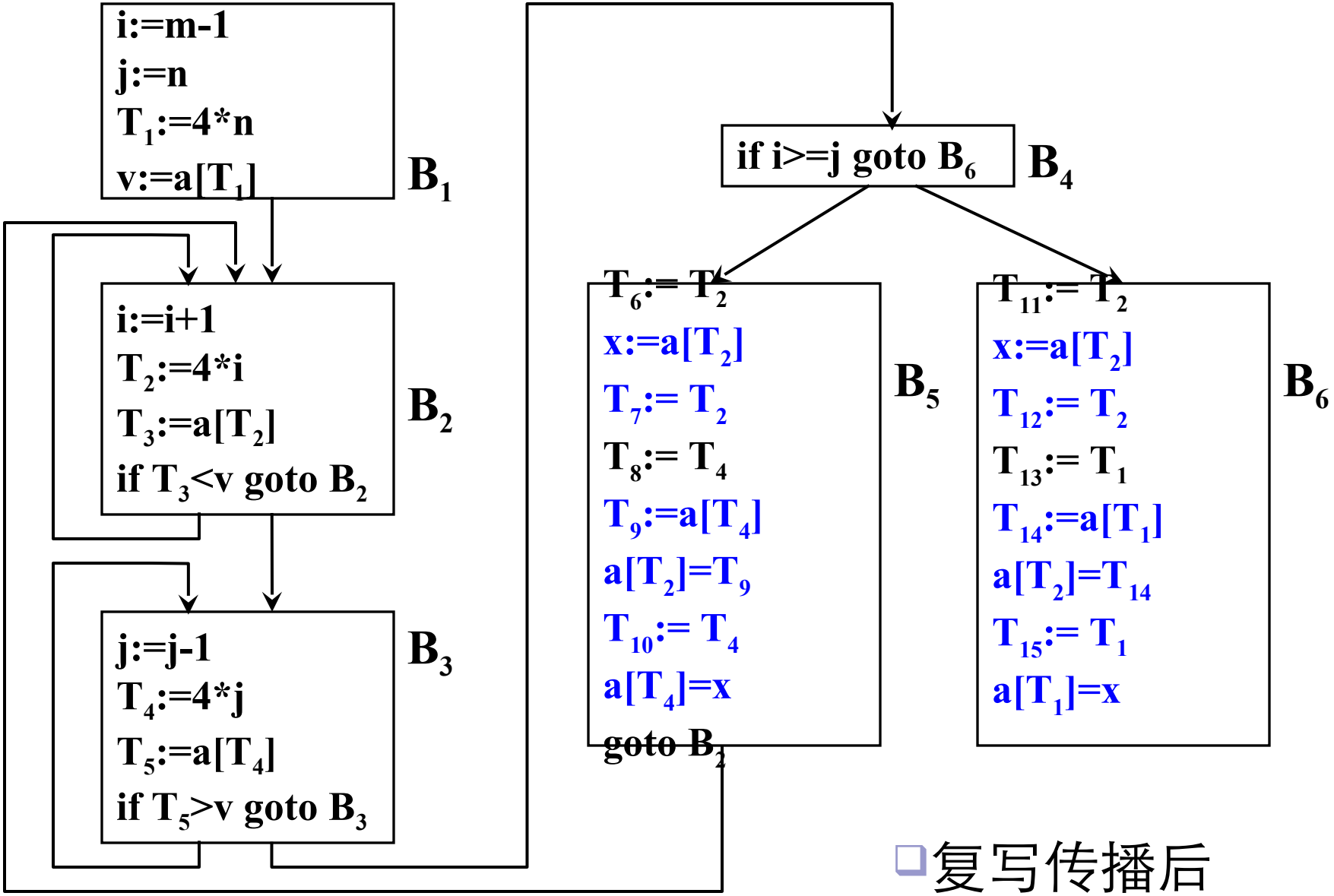
- 复写传播后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:= T2
x:=a[T2]
T7:= T2
T8:= T4
T9:=a[T4]
a[T2]=T9
T10:= T4
a[T4]=x
goto B2
B5
T11:= T2
x:=a[T2]
T12:= T2
T13:= T1
T14:=a[T13]
a[T2]=T14
T15:= T13
a[T15]=x
B6

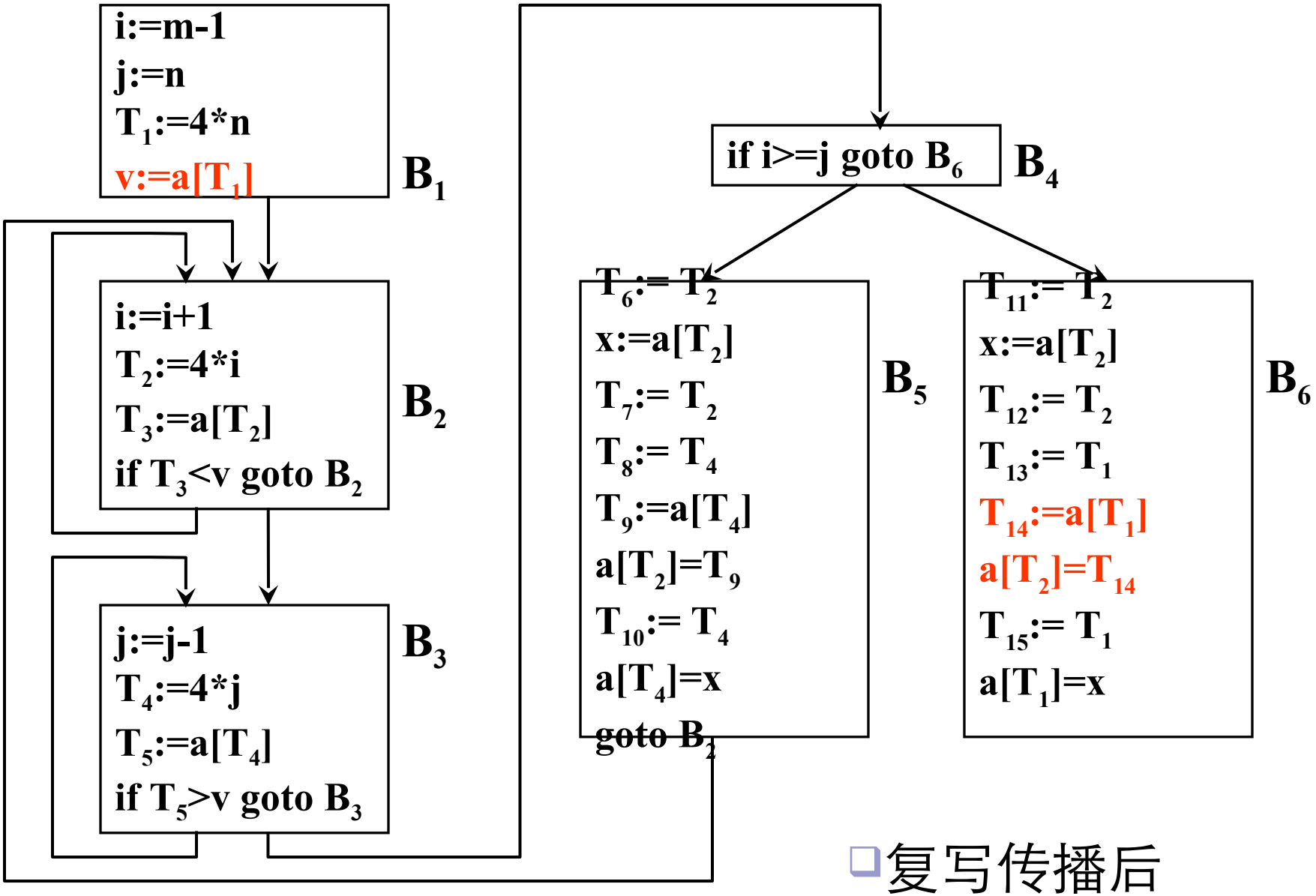
- 复写传播后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:= T2
x:=a[T2]
T7:= T2
T8:= T4
T9:=a[T4]
a[T2]=T9
T10:= T4
a[T4]=x
goto B2
B5
T11:= T2
x:=a[T2]
T12:= T2
T13:= T1
T14:=a[T13]
a[T2]=T14
T15:= T13
a[T15]=x
B6

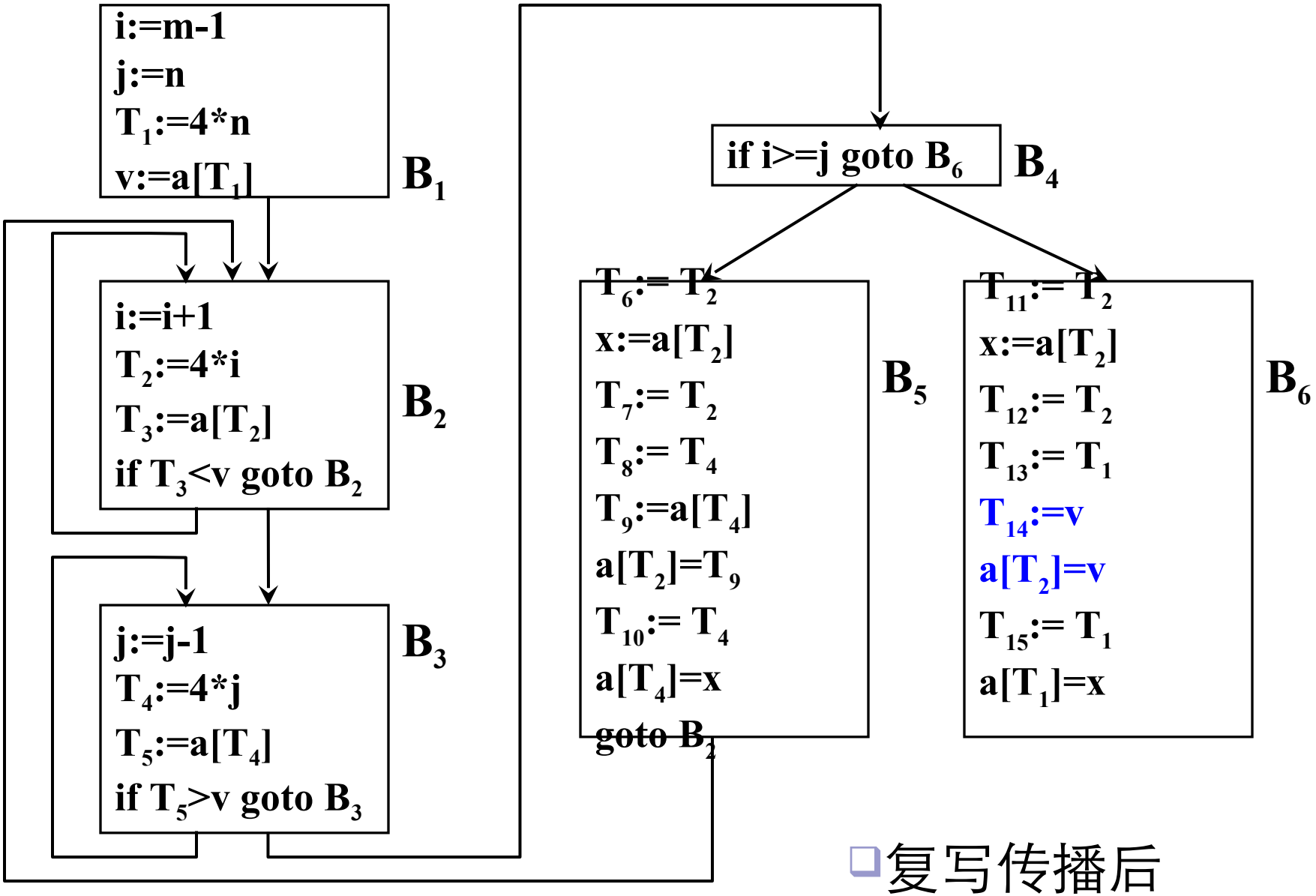
- 复写传播后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:= T2
x:=a[T2]
T7:= T2
T8:= T4
T9:=a[T4]
a[T2]=T9
T10:= T4
a[T4]=x
goto B2
B5
T11:= T2
x:=a[T2]
T12:= T2
T13:= T1
T14:=a[T1]
a[T2]=T14
T15:= T1
a[T1]=x
B6

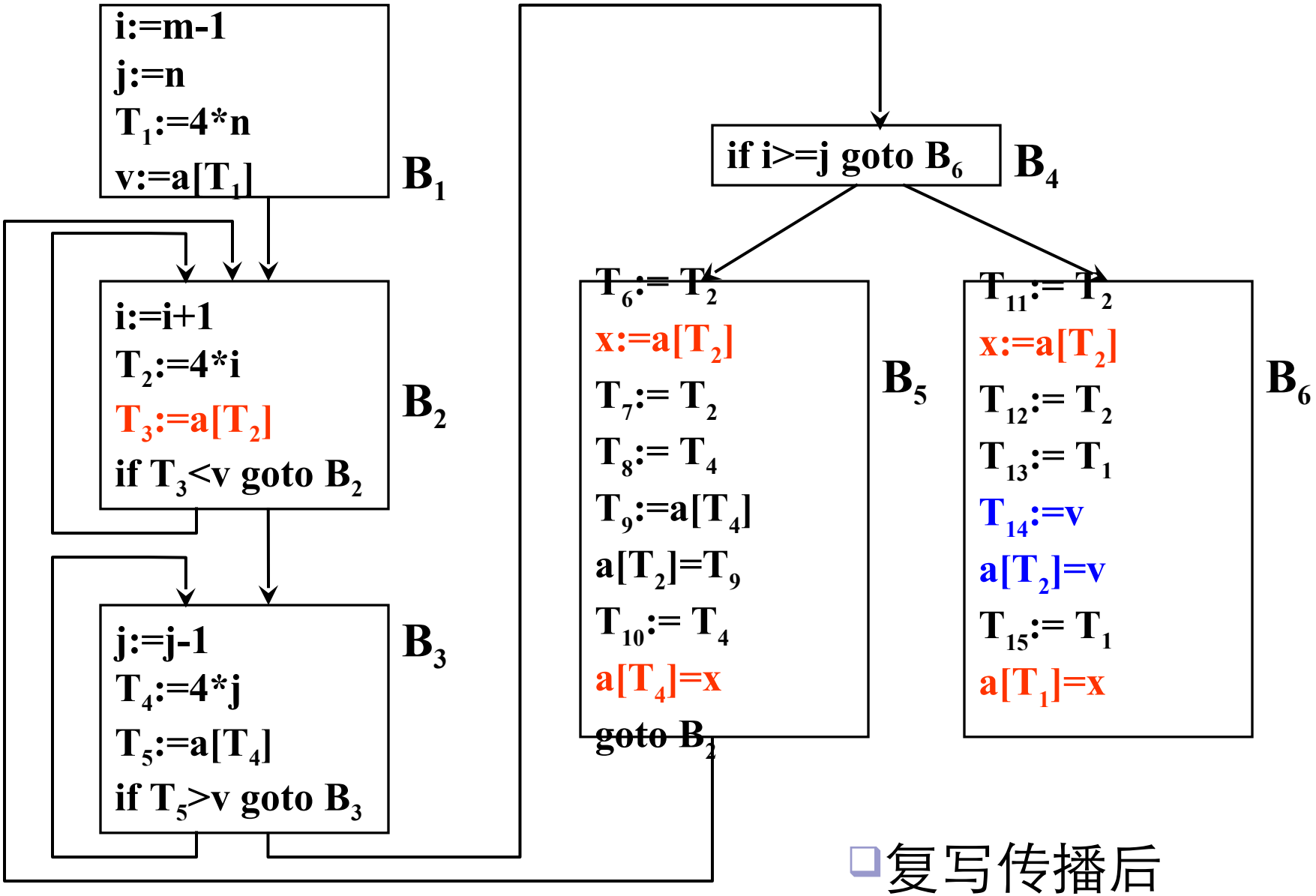
- 复写传播后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:= T2
x:=a[T2]
T7:= T2
T8:= T4
T9:=a[T4]
a[T2]=T9
T10:= T4
a[T4]=x
goto B2
B5
T11:= T2
x:=a[T2]
T12:= T2
T13:= T1
T14:=a[T1]
a[T2]=T14
T15:= T1
a[T1]=x
B6

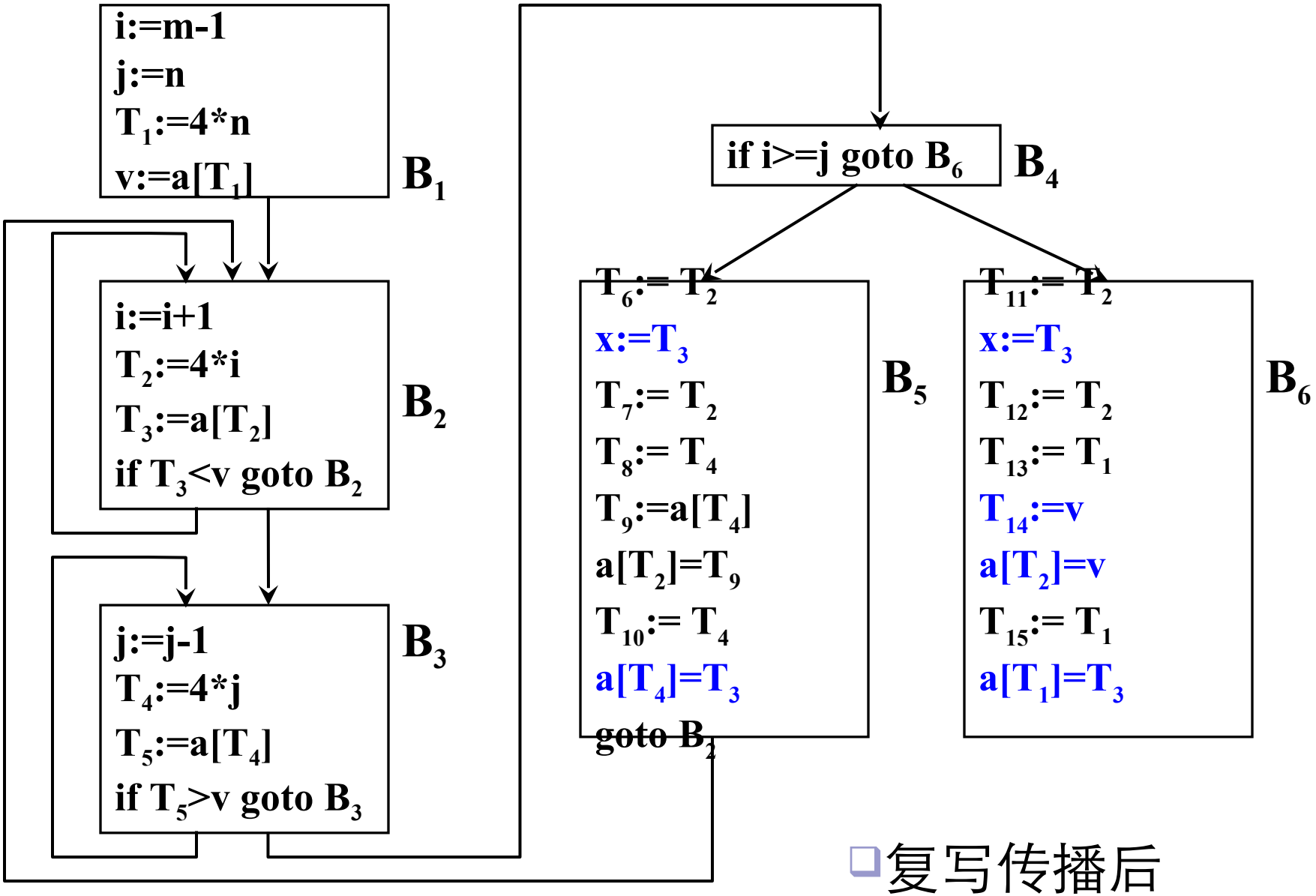
- 复写传播后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:= T2
x:=a[T2]
T7:= T2
T8:= T4
T9:=a[T4]
a[T2]=T9
T10:= T4
a[T4]=x
goto B2
B5
T11:= T2
x:=a[T2]
T12:= T2
T13:= T1
T14:=v
a[T2]=v
T15:= T1
a[T1]=x
B6

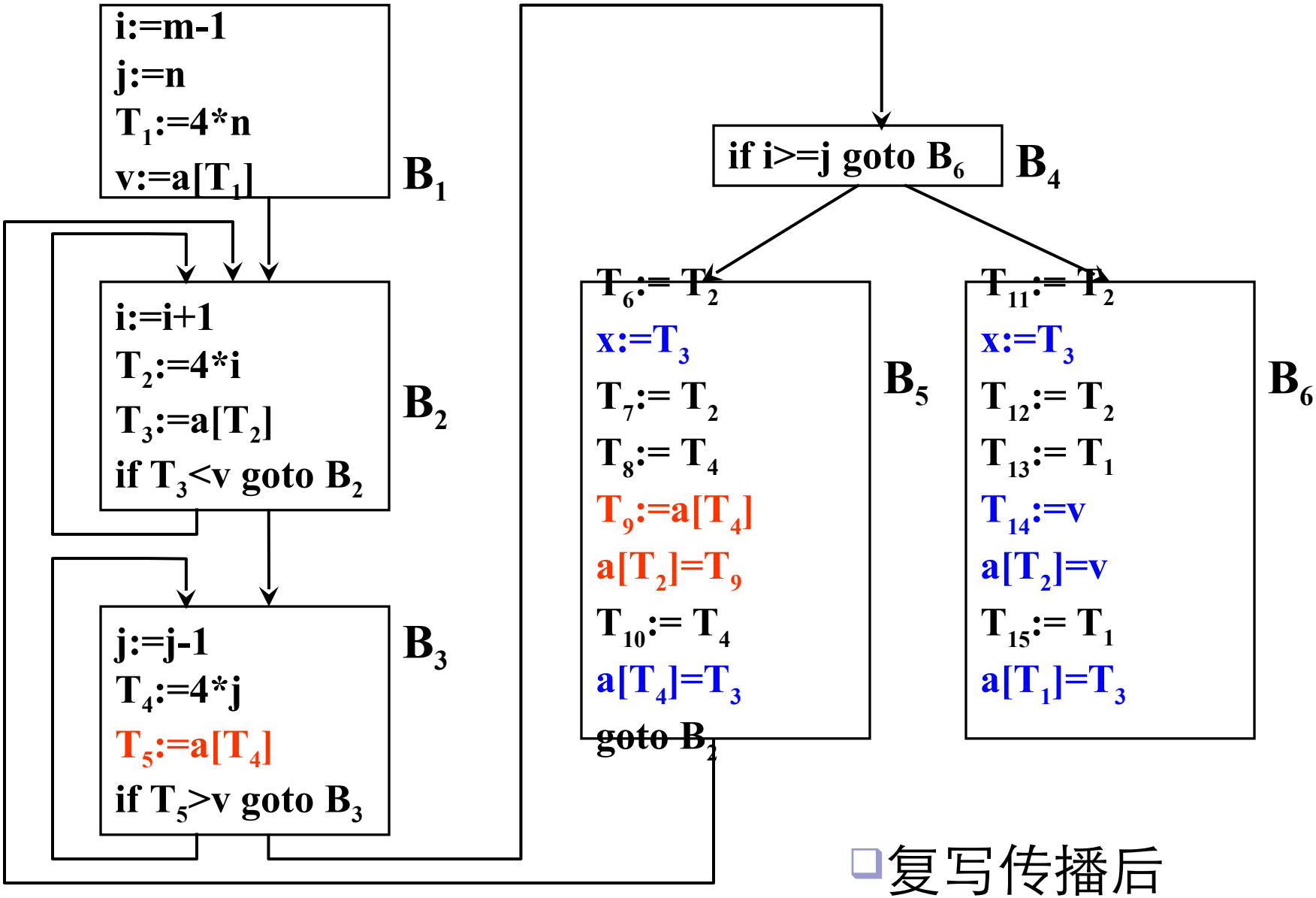
- 复写传播后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:= T2
x:=a[T2]
T7:= T2
T8:= T4
T9:=a[T4]
a[T2]=T9
T10:= T4
a[T4]=x
goto B2
B5
T11:= T2
x:=a[T2]
T12:= T2
T13:= T1
T14:=v
a[T2]=v
T15:= T1
a[T1]=x
B6

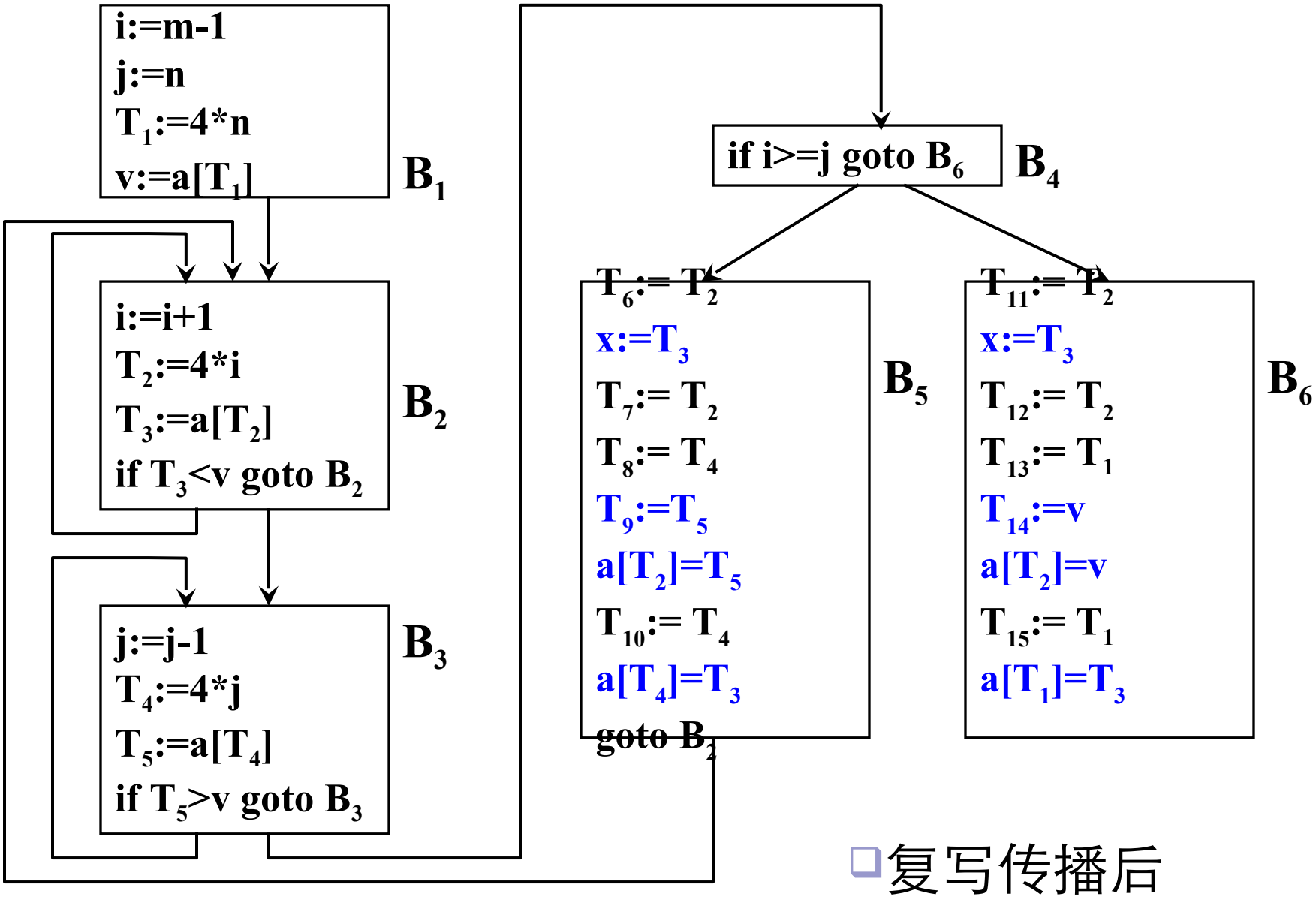
- 复写传播后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:= T2
x:=T3
T7:= T2
T8:= T4
T9:=a[T4]
a[T2]=T9
T10:= T4
a[T4]=T3
goto B2
B5
T11:= T2
x:=T3
T12:= T2
T13:= T1
T14:=v
a[T2]=v
T15:= T1
a[T1]=T3
B6

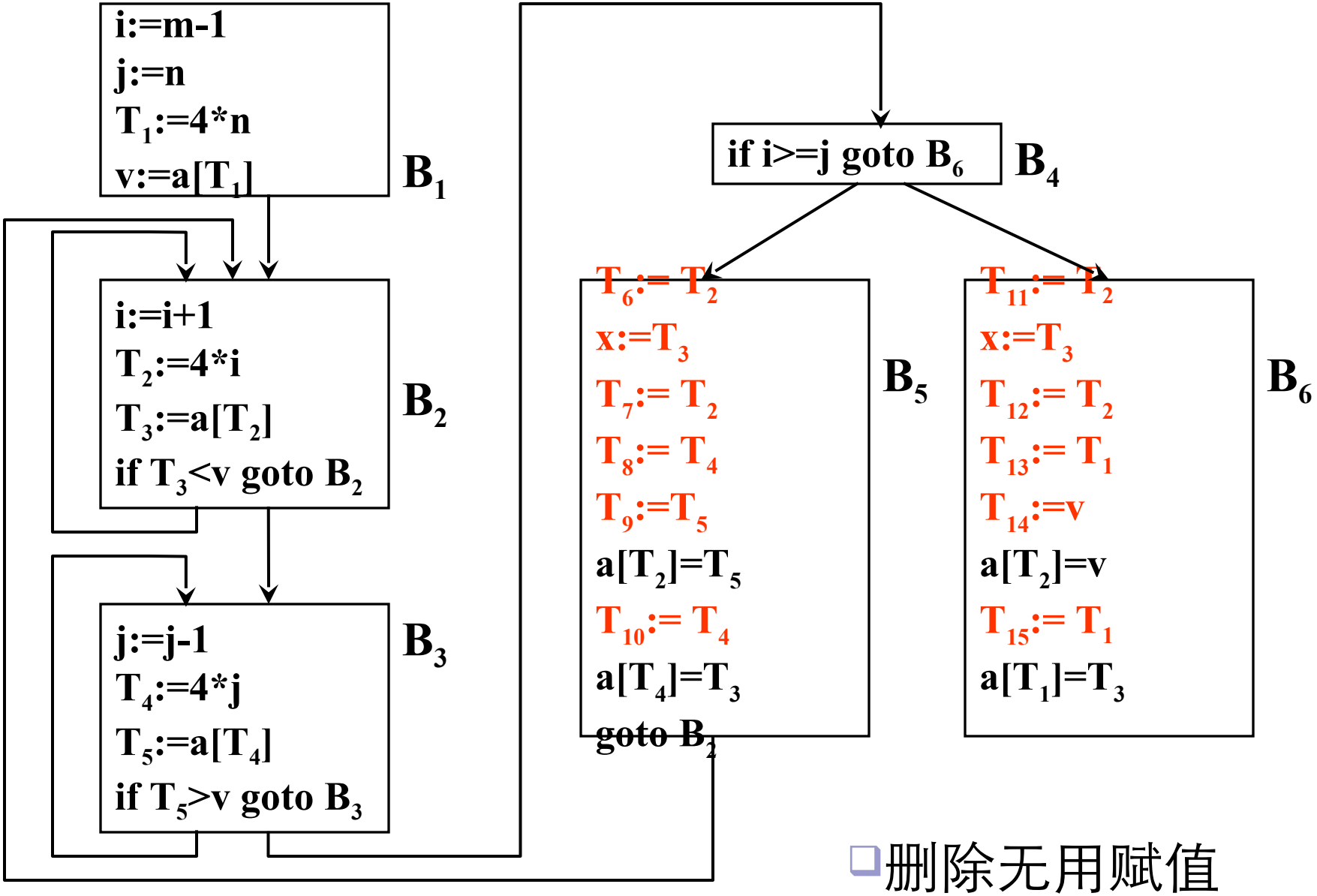
- 复写传播后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:= T2
x:=T3
T7:= T2
T8:= T4
T9:=a[T4]
a[T2]=T9
T10:= T4
a[T4]=T3
goto B2
B5
T11:= T2
x:=T3
T12:= T2
T13:= T1
T14:=v
a[T2]=v
T15:= T1
a[T1]=T3
B6

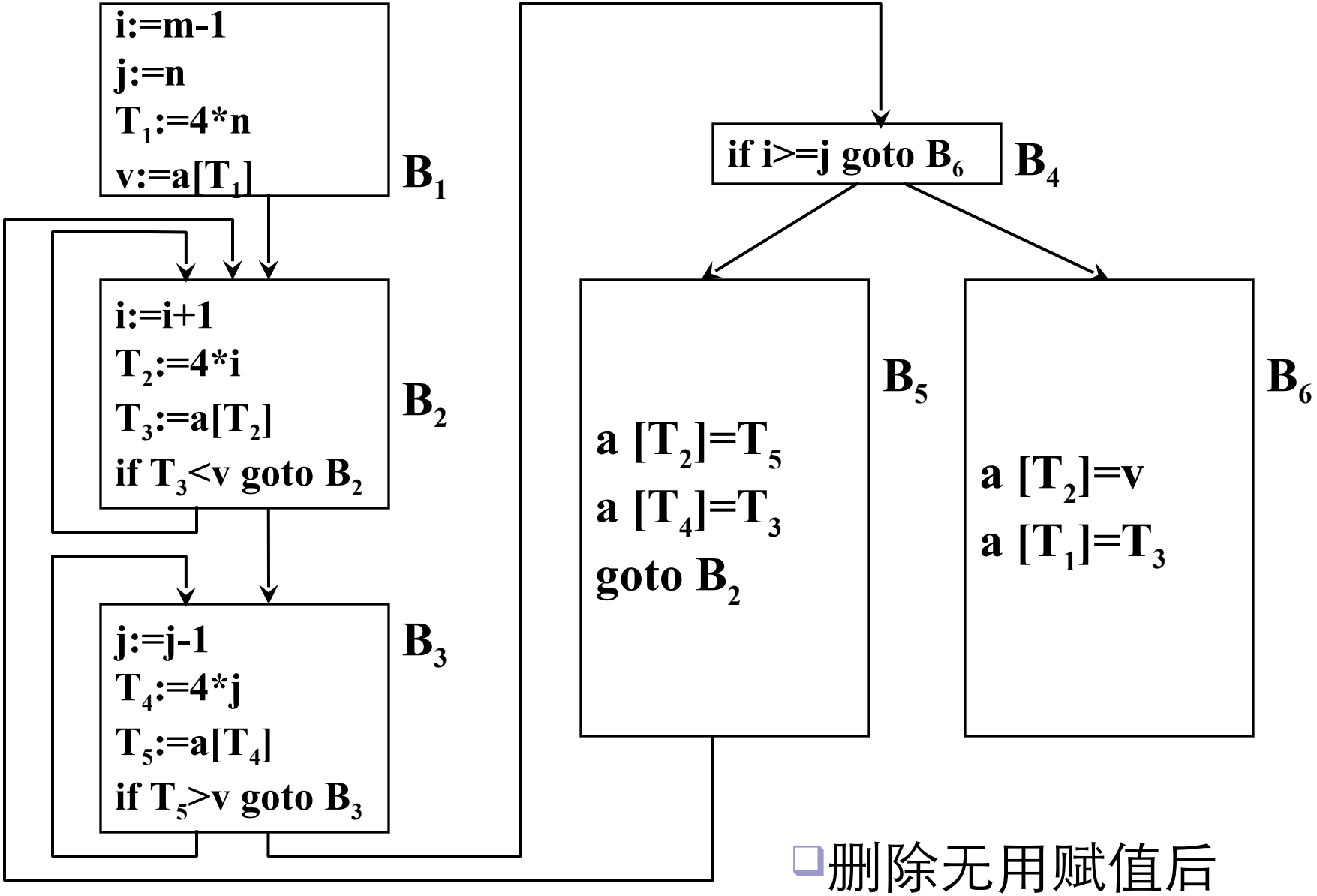
- 复写传播后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:= T2
x:=T3
T7:= T2
T8:= T4
T9:=T5
a[T2]=T5
T10:= T4
a[T4]=T3
goto B2
B5
T11:= T2
x:=T3
T12:= T2
T13:= T1
T14:=v
a[T2]=v
T15:= T1
a[T1]=T3
B6

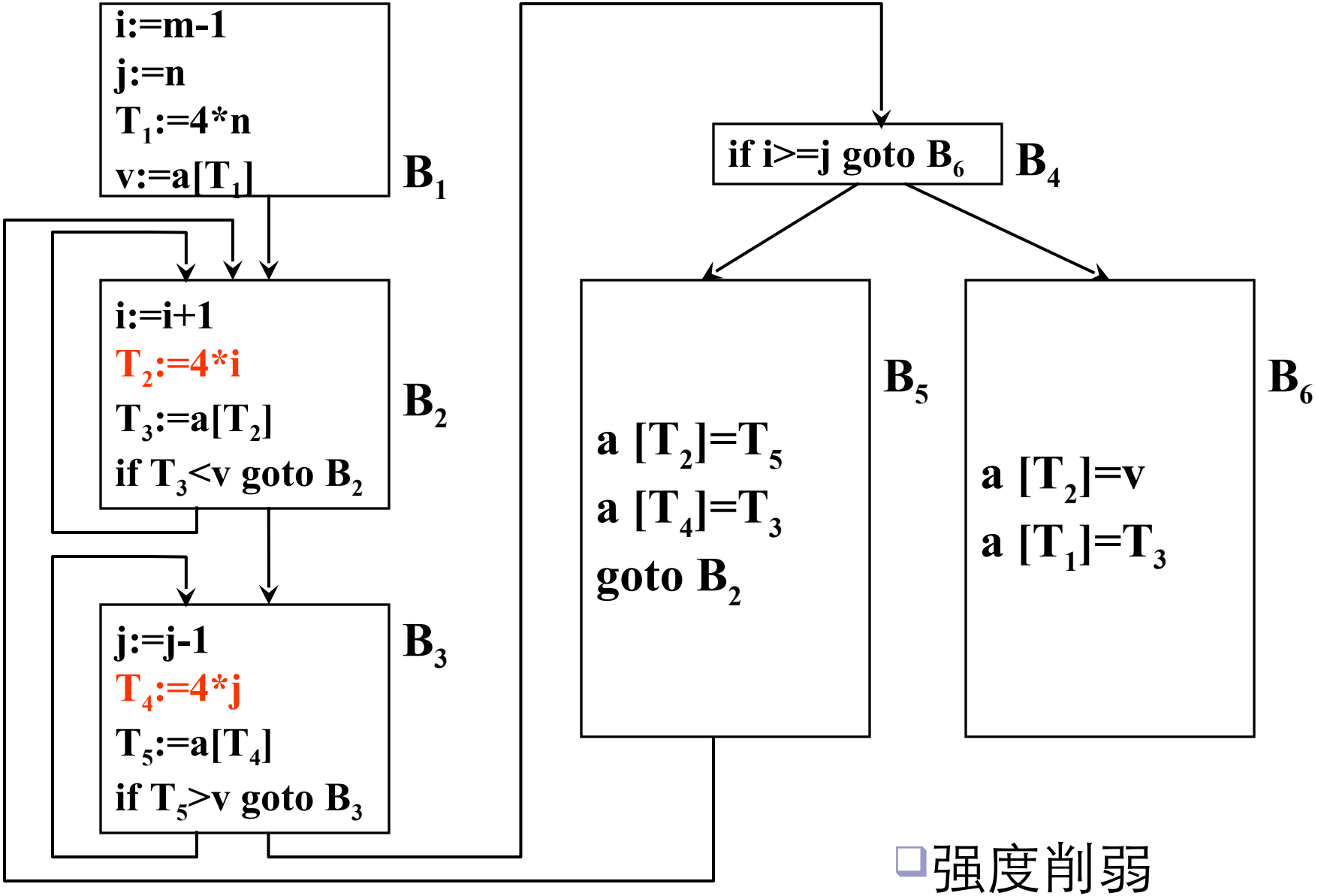
- 复写传播后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:= T2
x:=T3
T7:= T2
T8:= T4
T9:=T5
a[T2]=T5
T10:= T4
a[T4]=T3
goto B2
B5
T11:= T2
x:=T3
T12:= T2
T13:= T1
T14:=v
a[T2]=v
T15:= T1
a[T1]=T3
B6

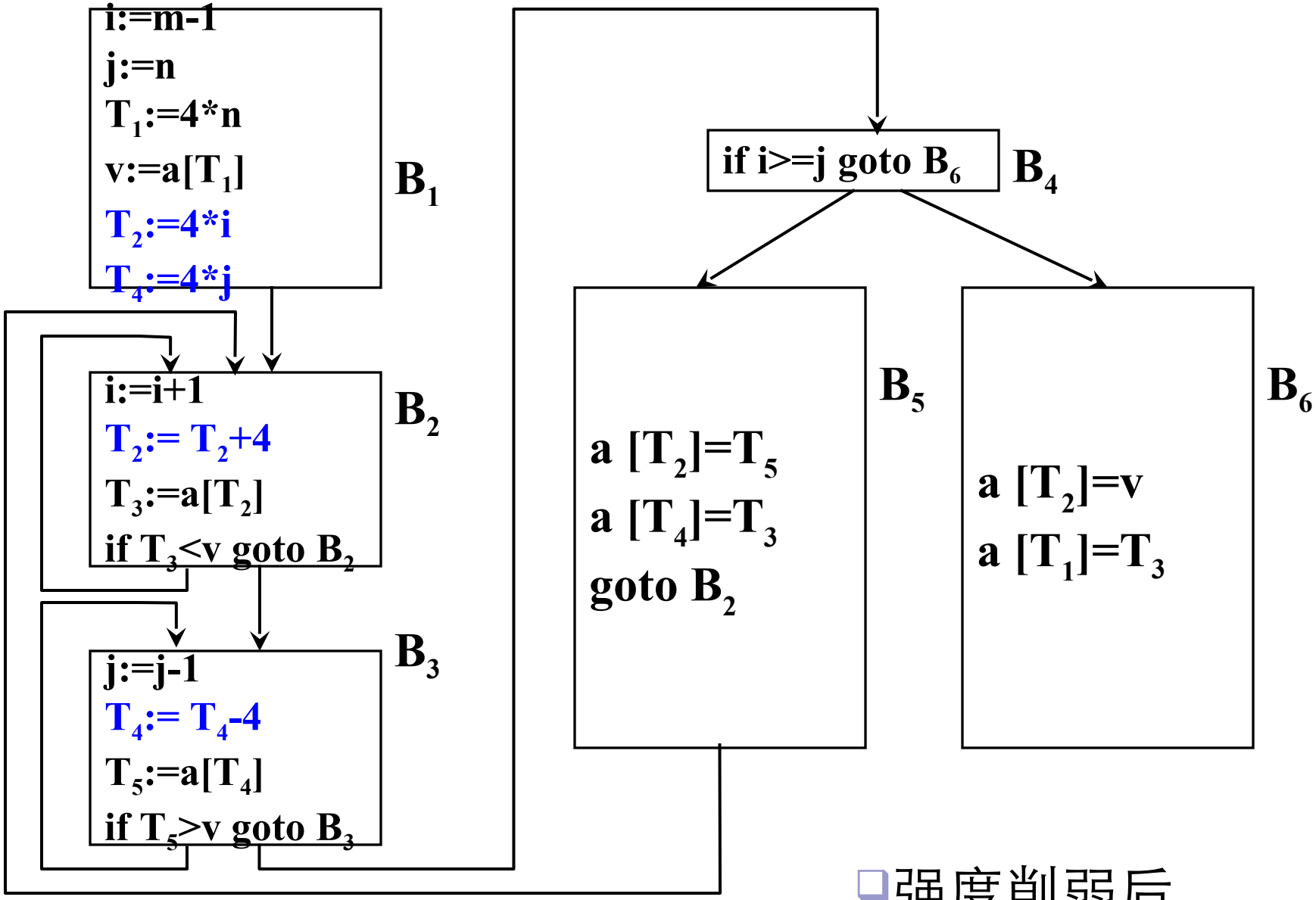
- 删除无用赋值


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
a [T2]=T5
a [T4]=T3
goto B2
B5
a [T2]=v
a [T1]=T3
B6

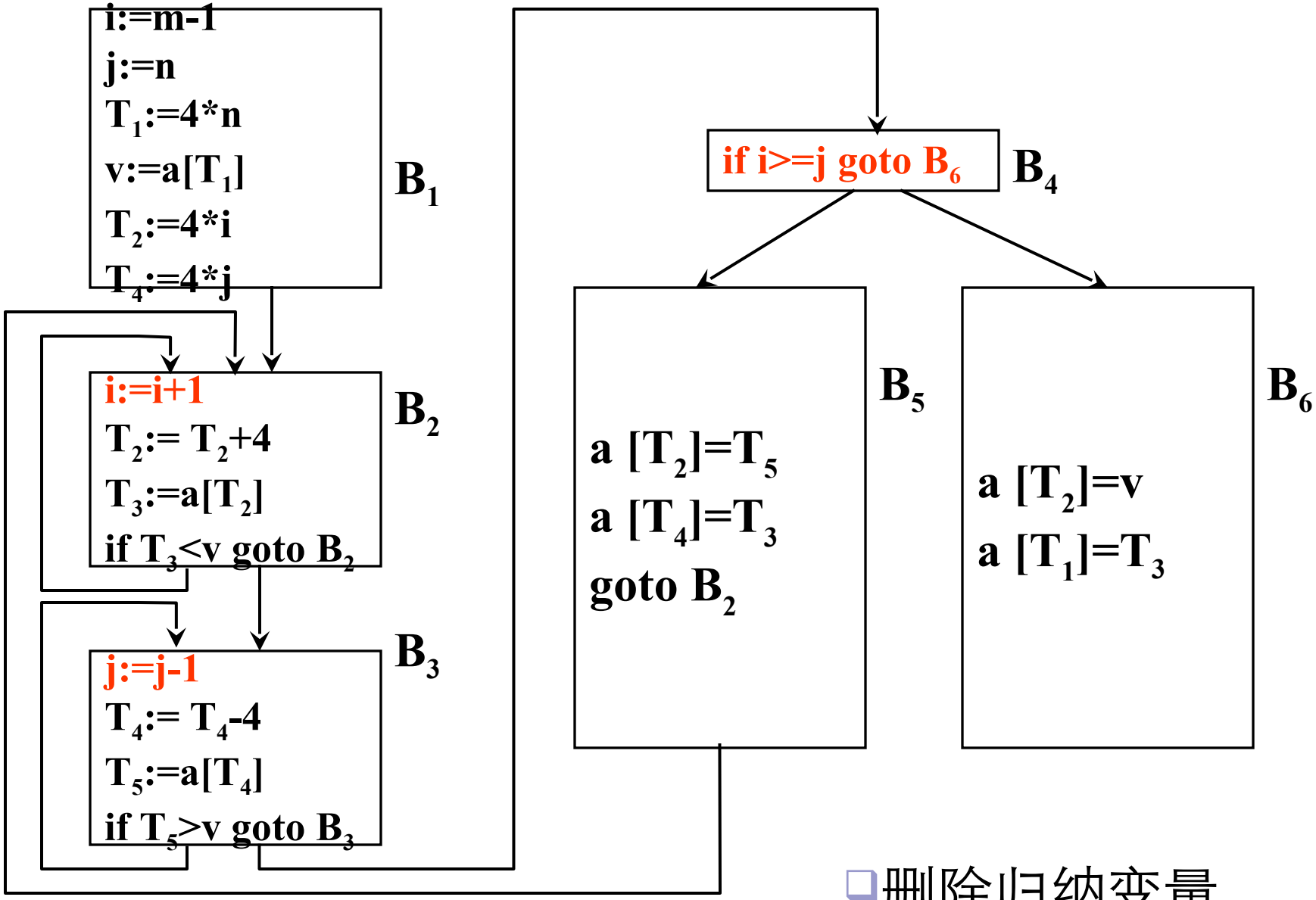
- 删除无用赋值后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
a [T2]=T5
a [T4]=T3
goto B2
B5
a [T2]=v
a [T1]=T3
B6

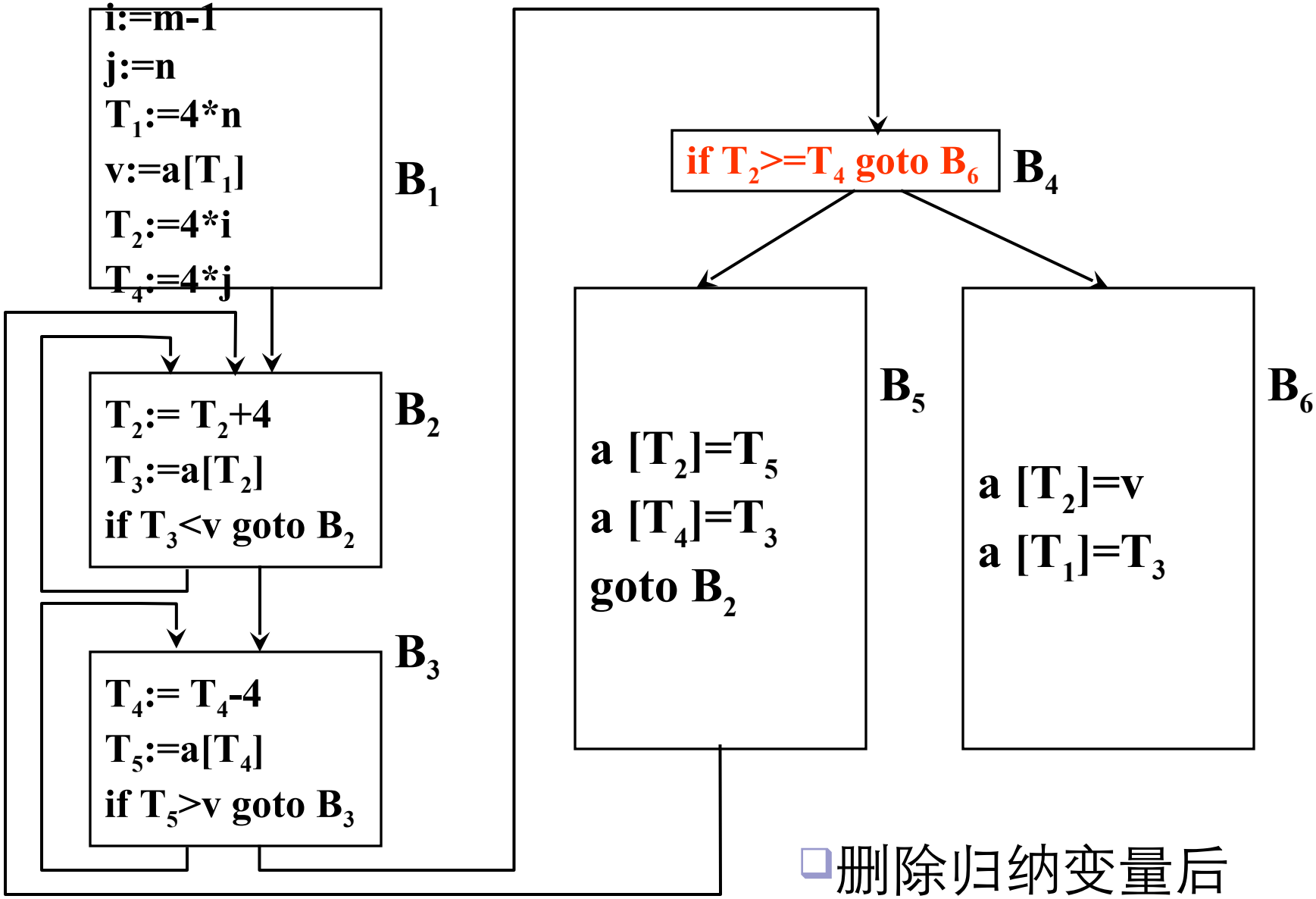
- 强度削弱


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
T2:=4*i
T4:=4*j
B1
i:=i+1
T2:= T2+4
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:= T4-4
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
a [T2]=T5
a [T4]=T3
goto B2
B5
a [T2]=v
a [T1]=T3
B6

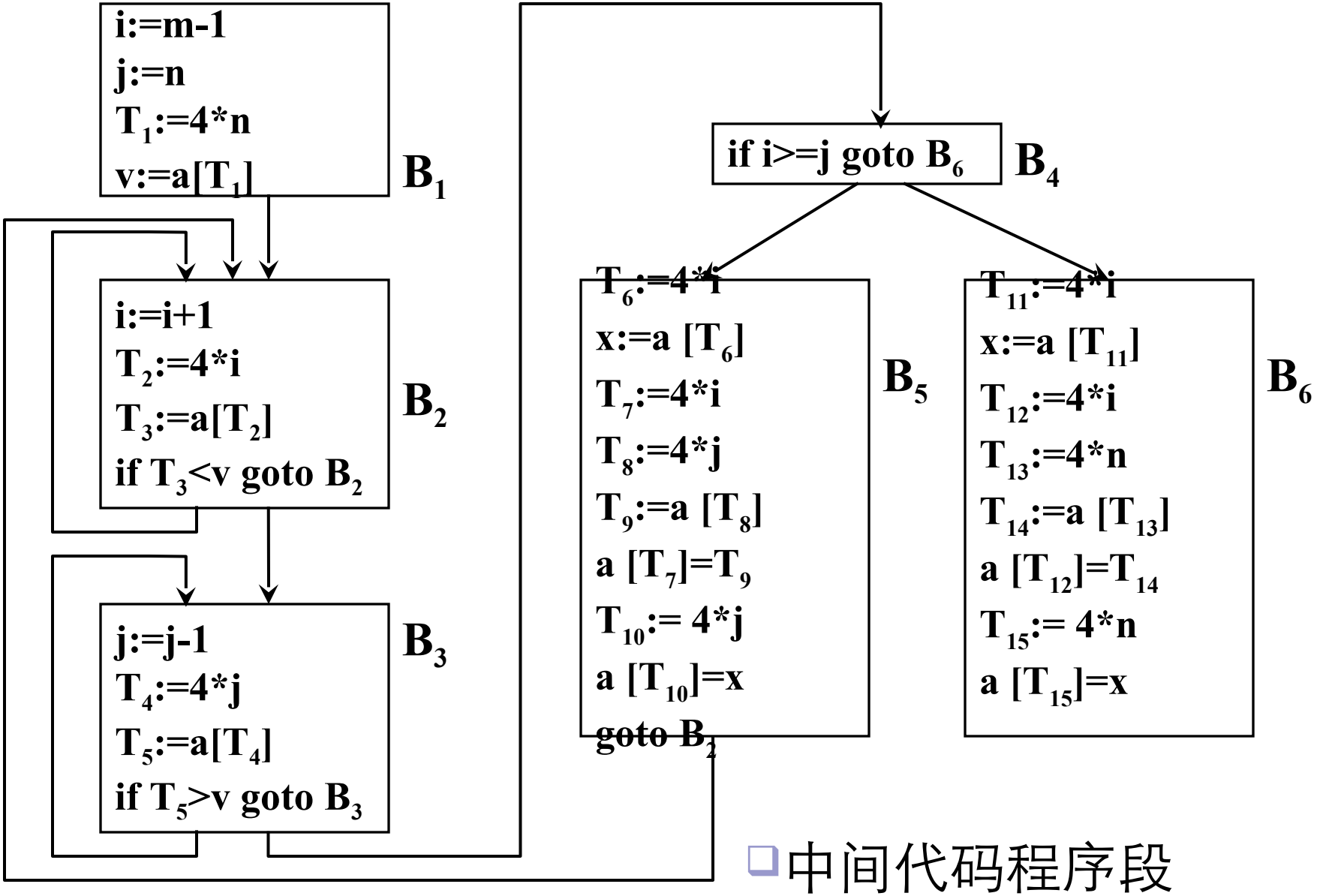
- 强度削弱后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
T2:=4*i
T4:=4*j
B1
i:=i+1
T2:= T2+4
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:= T4-4
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
a [T2]=T5
a [T4]=T3
goto B2
B5
a [T2]=v
a [T1]=T3
B6

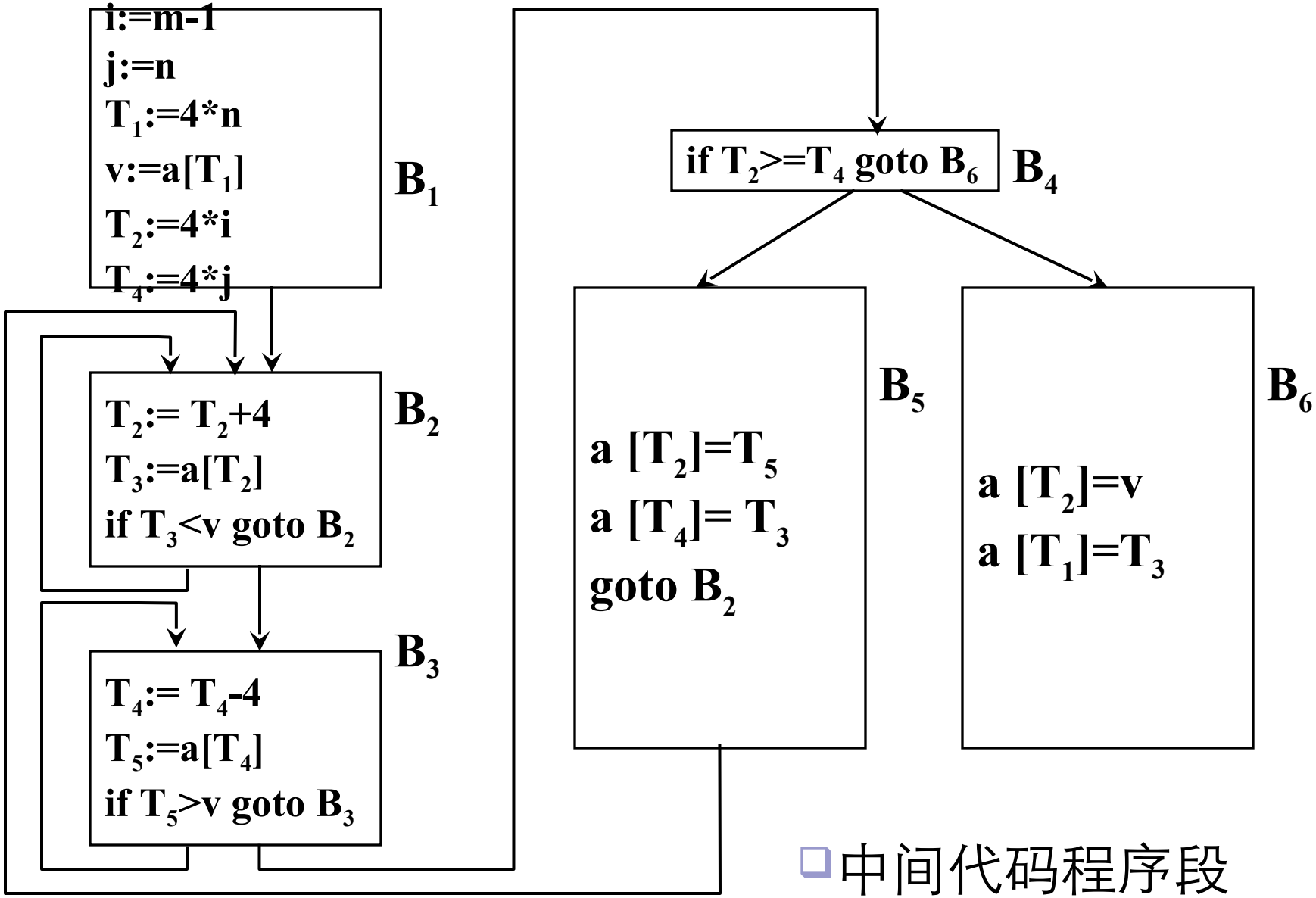
- 删除归纳变量


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
T2:=4*i
T4:=4*j
B1
T2:= T2+4
T3:=a[T2]
if T3<v goto B2
B2
T4:= T4-4
T5:=a[T4]
if T5>v goto B3
B3
if T2>=T4 goto B6
B4
a [T2]=T5
a [T4]=T3
goto B2
B5
a [T2]=v
a [T1]=T3
B6

- 删除归纳变量后


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
B1
i:=i+1
T2:=4*i
T3:=a[T2]
if T3<v goto B2
B2
j:=j-1
T4:=4*j
T5:=a[T4]
if T5>v goto B3
B3
if i>=j goto B6
B4
T6:=4*i
x:=a [T6]
T7:=4*i
T8:=4*j
T9:=a [T8]
a [T7]=T9
T10:= 4*j
a [T10]=x
goto B2
B5
T11:=4*i
x:=a [T11]
T12:=4*i
T13:=4*n
T14:=a [T13]
a [T12]=T14
T15:= 4*n
a [T15]=x
B6

- 中间代码程序段（优化前）


i:=m-1
j:=n
T1:=4*n
v:=a[T1]
T2:=4*i
T4:=4*j
B1
T2:= T2+4
T3:=a[T2]
if T3<v goto B2
B2
T4:= T4-4
T5:=a[T4]
if T5>v goto B3
B3
if T2>=T4 goto B6
B4
a [T2]=T5
a [T4]= T3
goto B2
B5
a [T2]=v
a [T1]=T3
B6

- 中间代码程序段（优化后）

第十章 优化

- 优化概述
- 局部优化
- 循环优化

10.2 局部优化

- 基本块：指程序中一顺序执行语句序列，其中只有一个入口和一个出口。入口就是其中第一个语句，出口就是其中最后一个语句。

入口 → $T_1:=a*a$
$T_2:=a*b$
$T_3:=2*T_2$
$T_4:=T_1+T_2$
出口 → $T_5:=b*b$
$T_6:=T_4+T_5$

- 如果一条三地址语句为 x:=y+z，则称对 x 定值并引用 y 和 z
- 基本块中的一个名字在程序中的某个给定点是活跃的，是指如果在程序中（包括在本基本块或在其它基本块中）它的值在该点以后被引用

10.2 局部优化

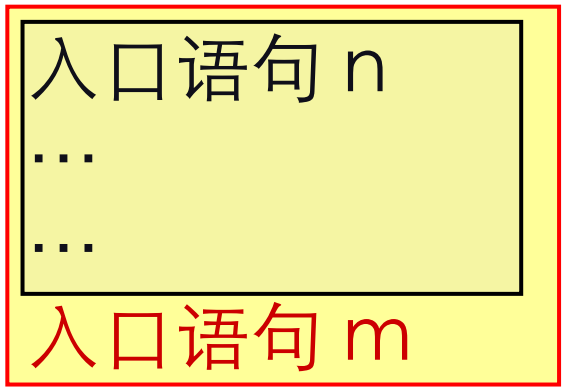
- 局限于基本块范围内的优化称为基本块内的优化，或称局部优化
- 如何将程序划分成基本块?
  - 基本块：指程序中一顺序执行语句序列，其中只有一个入口和一个出口。入口就是其中第一个语句，出口就是其中最后一个语句。

划分四元式程序为基本块的算法

1. 求出四元式程序中各个基本块的入口语句：
   1) 程序第一个语句，或
   2) 能由条件转移语句或无条件转移语句转移到的语句，或
   3) 紧跟在条件转移语句后面的语句

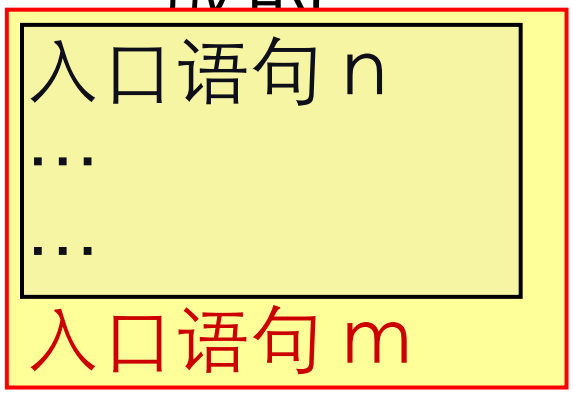
划分四元式程序为基本块的算法

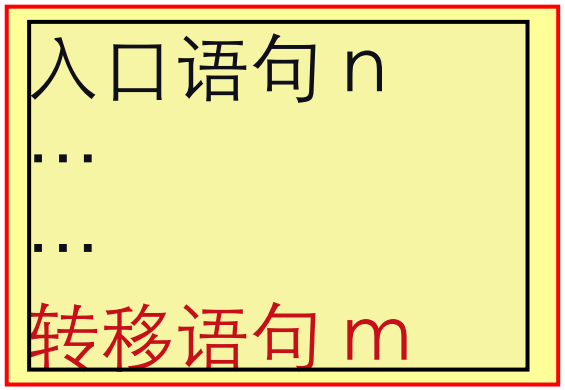
2. 对以上求出的每个入口语句，确定其所属的基本块。它是由该入口语句到下一入口语句（不包括该入口语句）之间的语句序列组成的。



划分四元式程序为基本块的算法

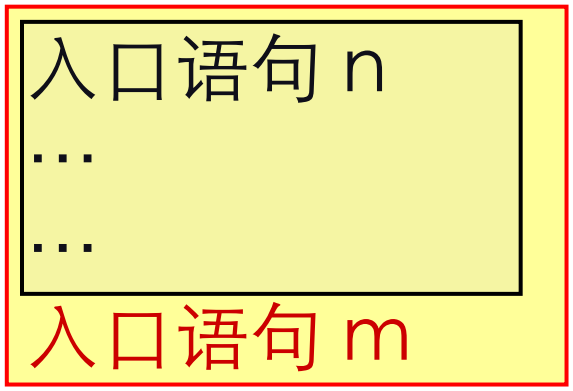
2. 对以上求出的每个入口语句，确定其所属的基本块。它是由该入口语句到下一入口语句（不包括该入口语句）、或到一转移语句（包括该转移语句）之间的语句序列组成的

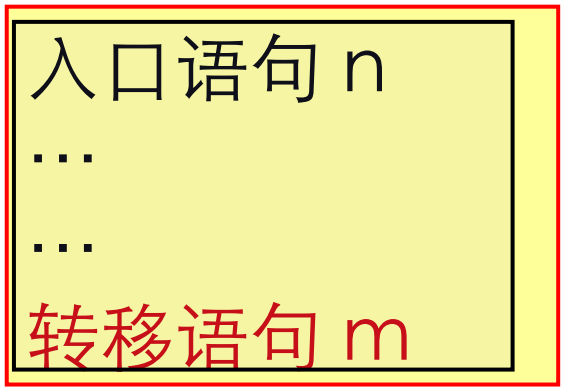


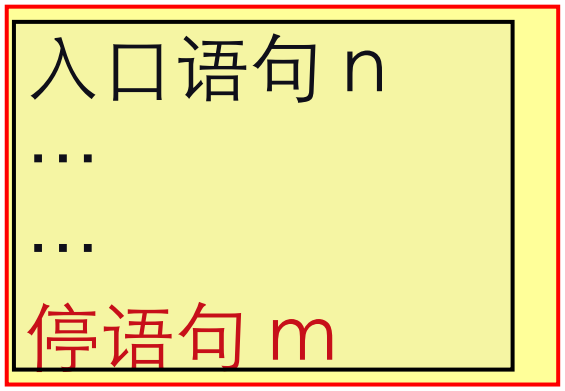


划分四元式程序为基本块的算法

2. 对以上求出的每个入口语句，确定其所属的基本块。它是由该入口语句到下一入口语句（不包括该入口语句）、或到一转移语句（包括该转移语句）、或一停语句（包括该停语句）之间的语句序列组成的。







划分四元式程序为基本块的算法

2. 对以上求出的每个入口语句，确定其所属的基本块。它是由该入口语句到下一入口语句（不包括该入口语句）、或到一转移语句（包括该转移语句）、或一停语句（包括该停语句）之间的语句序列组成的。

3. 凡未被纳入某一基本块中的语句，可以从程序中删除。

1. 求出四元式程序中各个基本块的入口语句：
   1) 程序第一个语句，或
   2) 能由条件转移语句或无条件转移语句转移到的语句，或
   3) 紧跟在条件转移语句后面的语句

- 例：划分基本块

```
(1)          read
(2)      read Y
(3)      R:=X mod Y
(4)      if  R=0 goto (8)
(5)      X:=Y
(6)      Y:=R
(7)      goto (3)
(8)      write Y
(9)      halt
```

1. 求出四元式程序中各个基本块的入口语句：
   1) 程序第一个语句，或
   2) 能由条件转移语句或无条件转移语句转移到的语句，或
   3) 紧跟在条件转移语句后面的语句

- 例：划分基本块

(1) rea
(2) read Y
(3) R:=X mod Y
(4) if R=0 goto (8)
(5) X:=Y
(6) Y:=R
(7) goto (3)
(8) write Y
(9) halt

1. 求出四元式程序中各个基本块的入口语句：
   1) 程序第一个语句，或
   2) 能由条件转移语句或无条件转移语句转移到的语句，或
   3) 紧跟在条件转移语句后面的语句

- 例：划分基本块

```
(1)          read
(2)          read Y
(3)          R:=X mod Y
(4)          if  R=0 goto (8)
(5)          X:=Y
(6)          Y:=R
(7)          goto (3)
(8)          write Y
(9)          halt
```

1. 求出四元式程序中各个基本块的入口语句：
   1) 程序第一个语句，或
   2) 能由条件转移语句或无条件转移语句转移到的语句，或
   3) 紧跟在条件转移语句后面的语句

- 例：划分基本块

```
(1)          read
(2)          read Y
(3)          R:=X mod Y
(4)          if  R=0 goto (8)
(5)          X:=Y
(6)          Y:=R
(7)          goto (3)
(8)          write Y
(9)          halt
```

1. 求出四元式程序中各个基本块的入口语句：
   1) 程序第一个语句，或
   2) 能由条件转移语句或无条件转移语句转移到的语句，或
   3) 紧跟在条件转移语句后面的语句

- 例：划分基本块

```
(1)          read
(2)          read Y
(3)          R:=X mod Y
(4)          if  R=0 goto (8)
(5)          X:=Y
(6)          Y:=R
(7)          goto (3)
(8)          write Y
(9)          halt
```

- 例：划分基本块

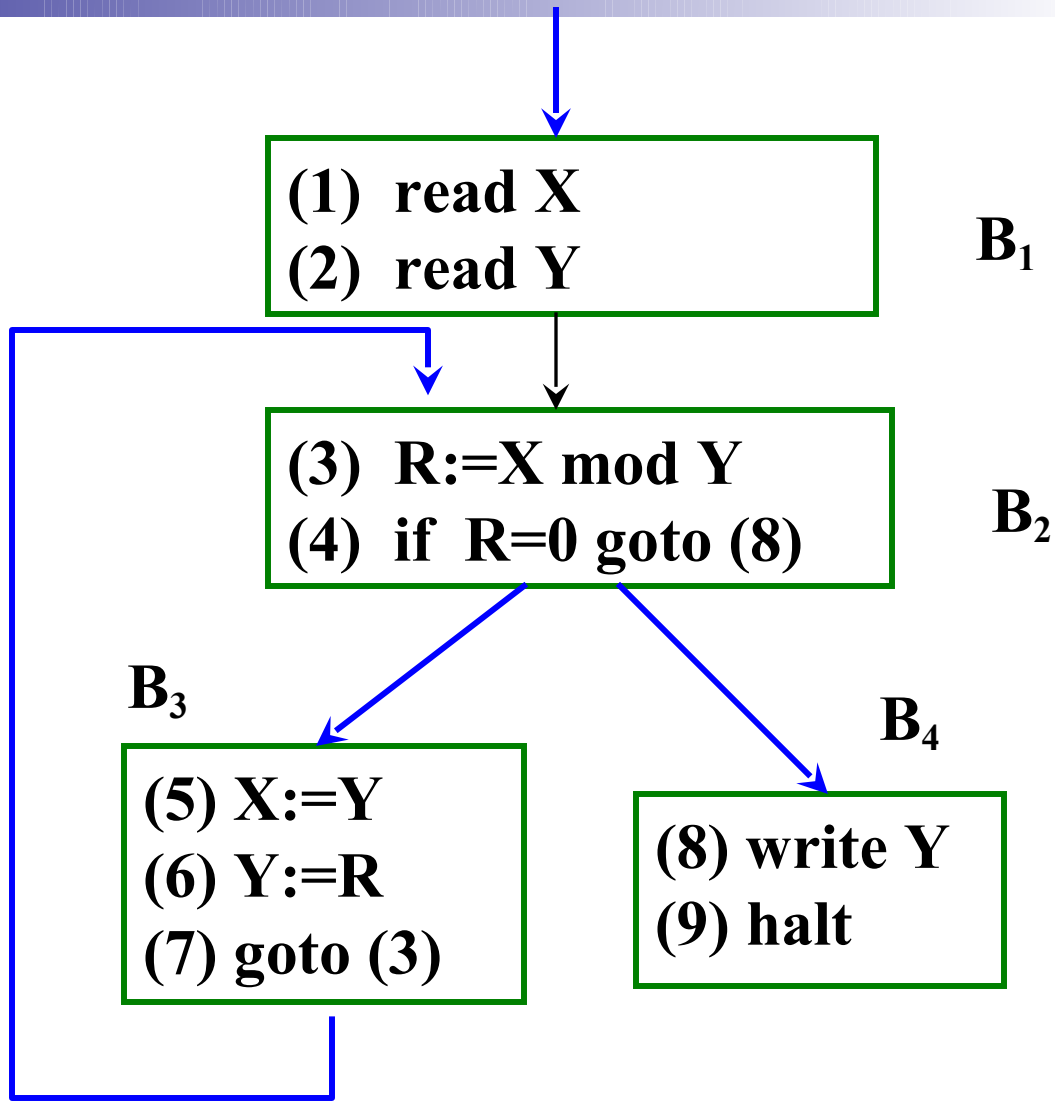
```
(1)        read X
(2)        read Y
(3)        R:=X mod Y
(4)        if  R=0 goto (8)
(5)        X:=Y
(6)        Y:=R
(7)        goto (3)
(8)        write Y
(9)        halt
```

2. 对以上求出的每个入口语句，确定其所属的基本块。它是由该入口语句到下一入口语句（不包括该入口语句）、或到一转移语句（包括该转移语句）、或一停语句（包括该停语句）之间的语句序列组成的。

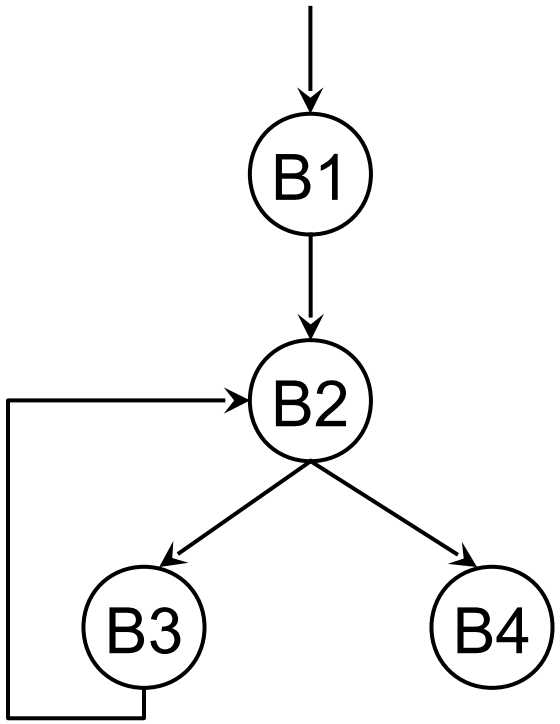
流图

- 每个流图以基本块为结点
- 如果一个结点的基本块的入口语句是程序的第一条语句，则称此结点为首结点
- 如果在某个执行顺序中，基本块 B_2 紧接在基本块 B_1 之后执行，则从 B_1 到 B_2 有一条有向边。即，如果
  - 有一个条件或无条件转移语句从 B_1 的最后一条语句转移到 B_2 的第一条语句；或者
  - 在程序的序列中， B_2 紧接在 B_1 的后面，并且 B_1 的最后一条语句不是一个无条件转移语句。我们就说 B_1 是 B_2 的前驱， B_2 是 B_1 的后继

```
(1)  read X
(2)  read Y
(3)  R:=X mod Y
(4)  if  R=0 goto (8)
(5)  X:=Y
(6)  Y:=R
(7)  goto (3)
(8)  write Y
(9)  halt
```



(1)	read X	B1
(2)	read Y	
(3)	R:=X mod Y	B2
(4)	if R=0 goto (8)	
(5)	X:=Y	B3
(6)	Y:=R	
(7)	goto (3)	
(8)	write Y	B4
(9)	halt	



小结

- 优化的基本概念
- 局部优化
  - 基本块划分算法

作业

- P306-1， 2