递归函数转非递归

胡船长

初航我带你，远航靠自己

本期内容

一. 系统栈模拟法

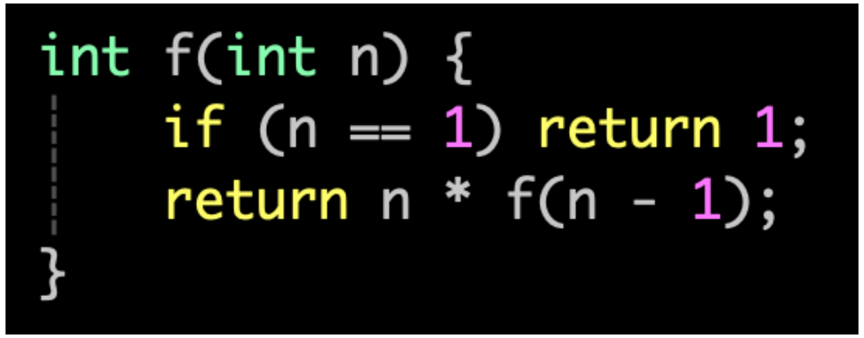
1. 系统栈模拟法原理讲解
2. 练习1：阶乘函数转非递归
3. 练习2：中序遍历转非递归
4. 练习3：快速排序转非递归

二. 拓扑序分解法

1. 拓扑序知识讲解
2. 练习4：HZOJ-641-拓扑排序
3. 练习5：HZOJ-636-旅行计划
4. 练习6：归并排序转非递归

一. 系统栈模拟法

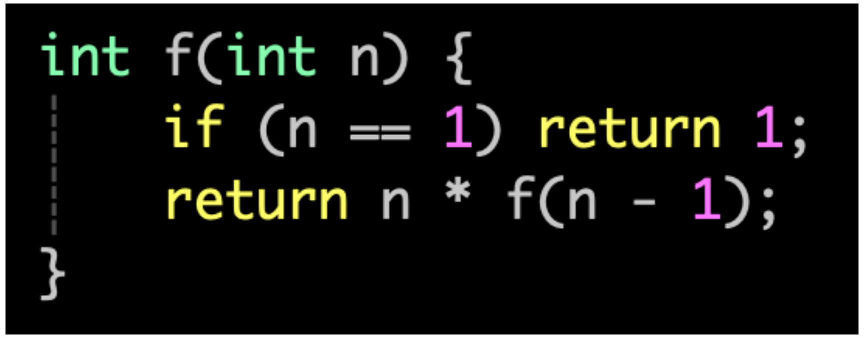
用栈模拟递归函数



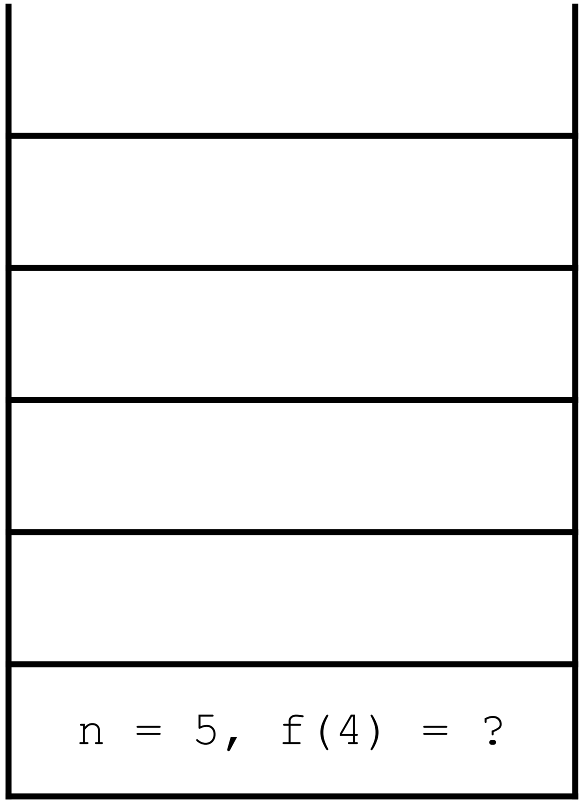
```
int f(int n) {
    if (n == 1) return 1;
    return n * f(n - 1);
}
```

系统栈

用栈模拟递归函数

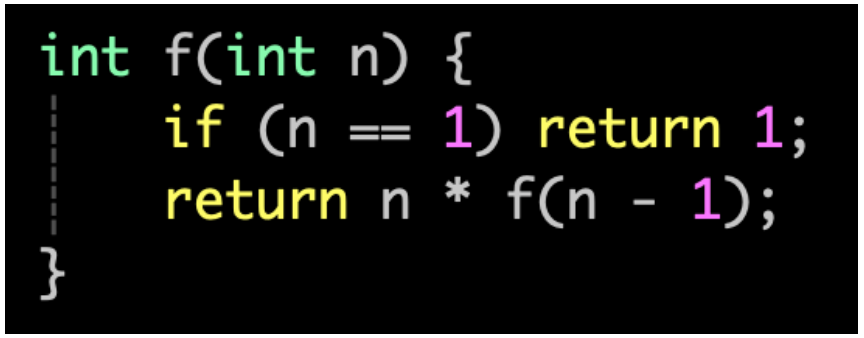


```
int f(int n) {
    if (n == 1) return 1;
    return n * f(n - 1);
}
```

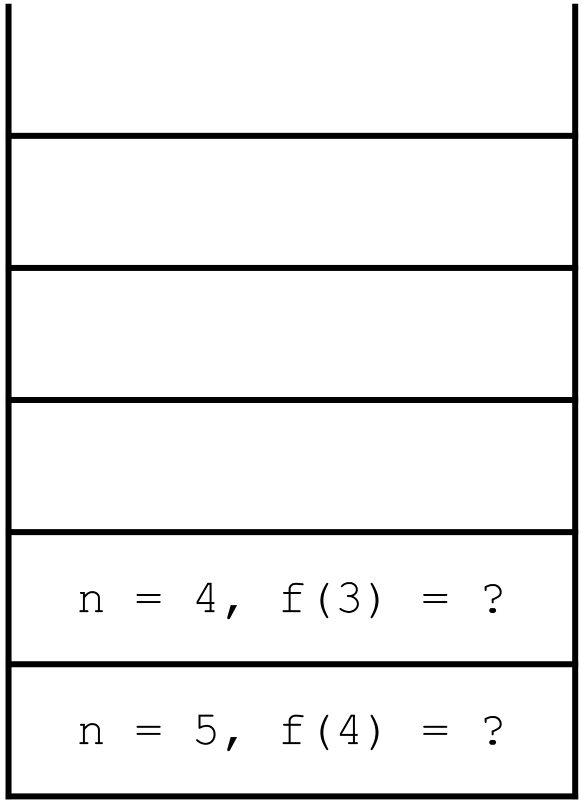


系统栈

用栈模拟递归函数

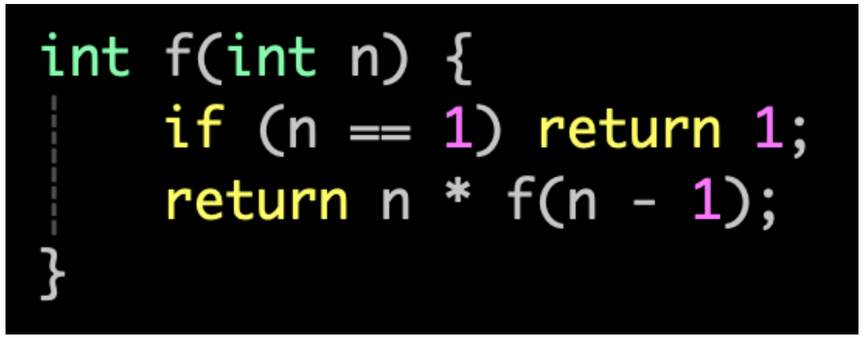


```
int f(int n) {
    if (n == 1) return 1;
    return n * f(n - 1);
}
```

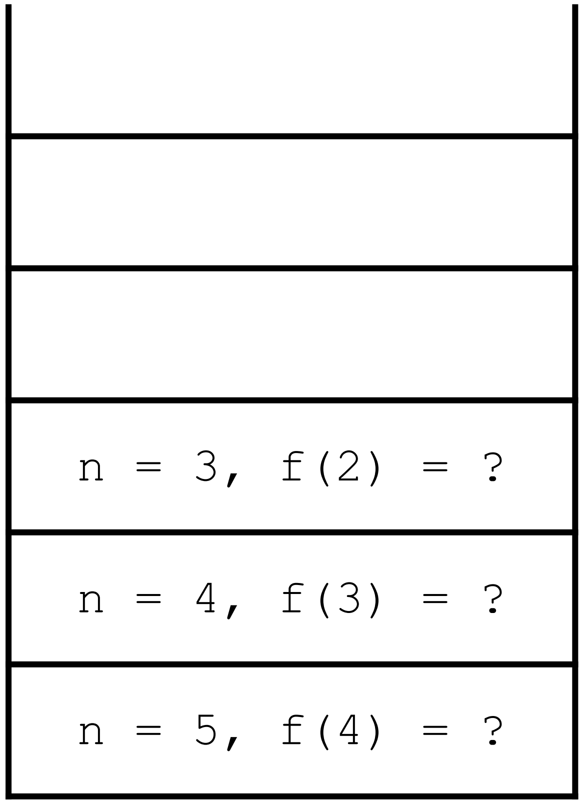


系统栈

用栈模拟递归函数

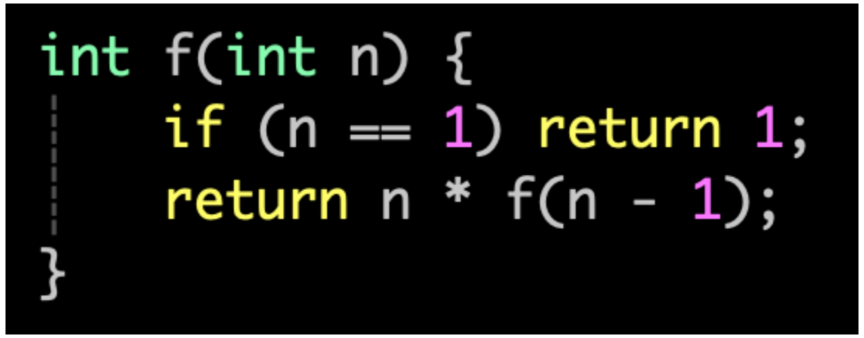


```
int f(int n) {
    if (n == 1) return 1;
    return n * f(n - 1);
}
```

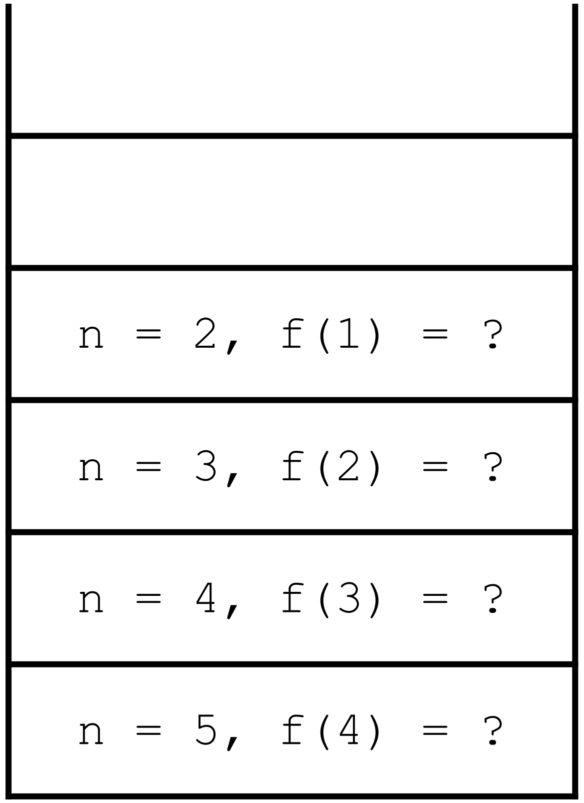


系统栈

用栈模拟递归函数

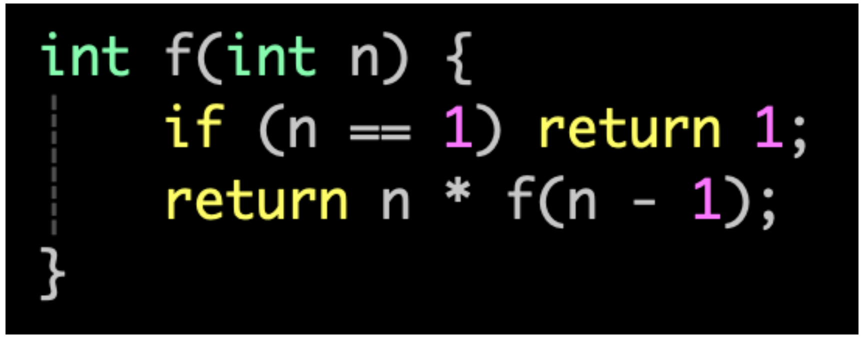


```
int f(int n) {
    if (n == 1) return 1;
    return n * f(n - 1);
}
```

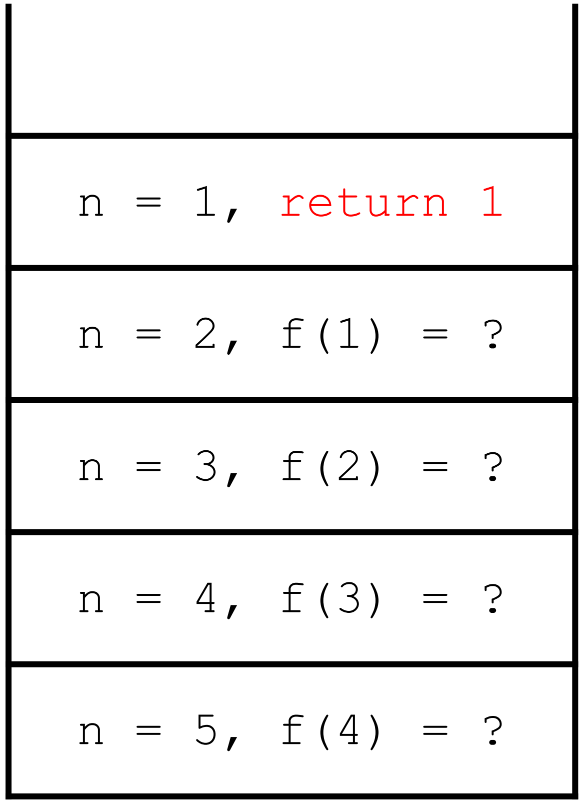


系统栈

用栈模拟递归函数

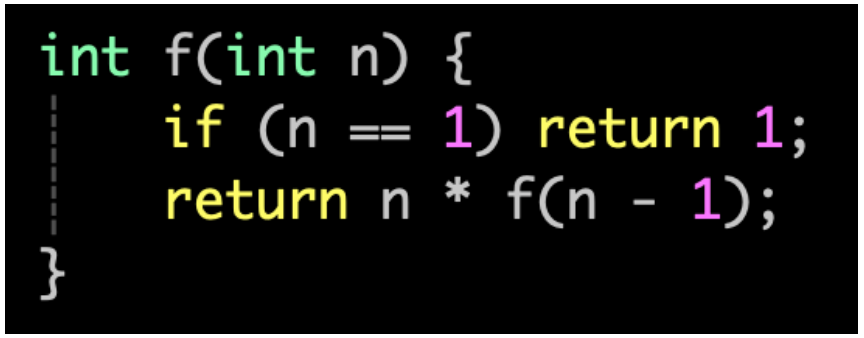


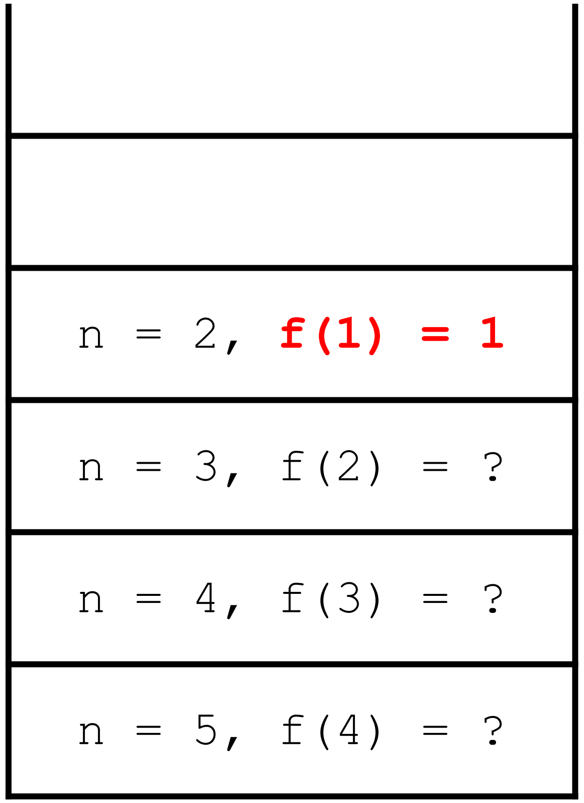
```
int f(int n) {
    if (n == 1) return 1;
    return n * f(n - 1);
}
```



系统栈

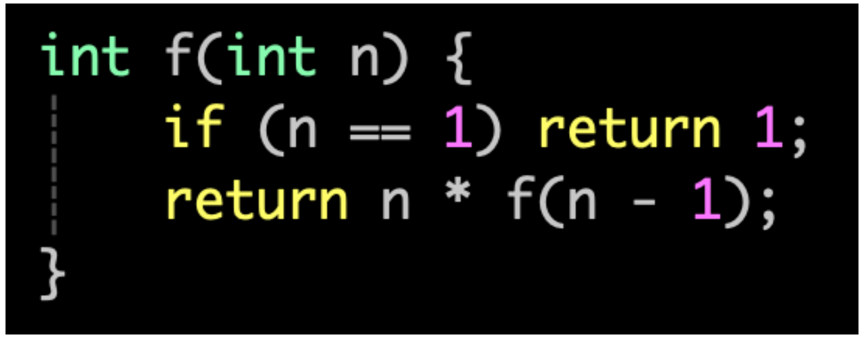
用栈模拟递归函数


```
int f(int n) {
    if (n == 1) return 1;
    return n * f(n - 1);
}
```

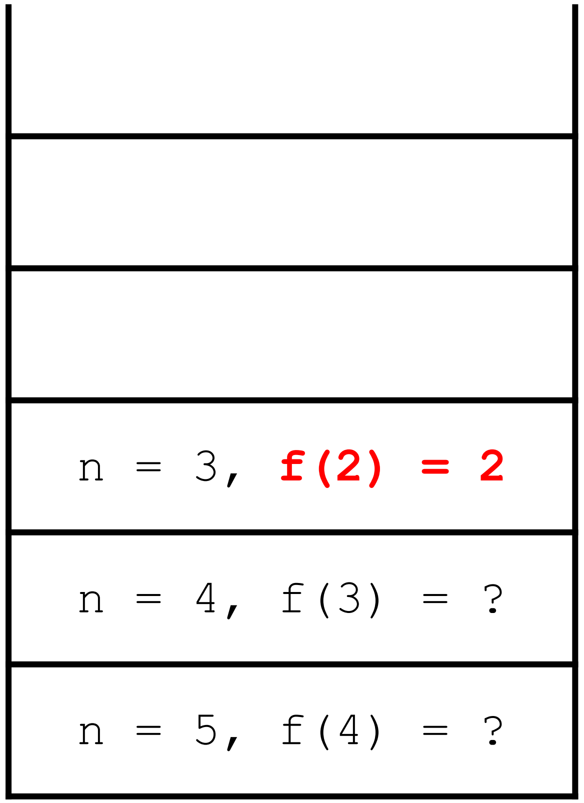


系统栈

用栈模拟递归函数

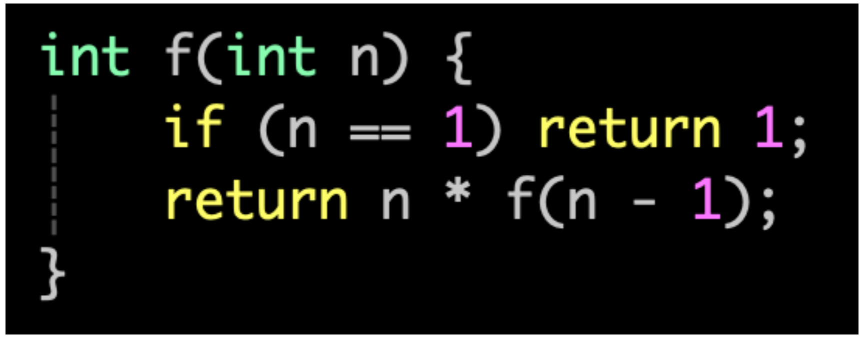


```
int f(int n) {
    if (n == 1) return 1;
    return n * f(n - 1);
}
```

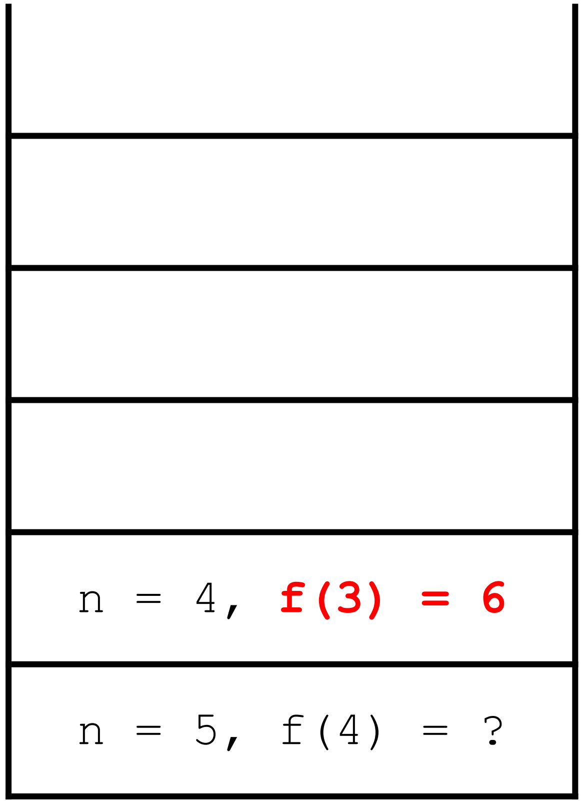


系统栈

用栈模拟递归函数

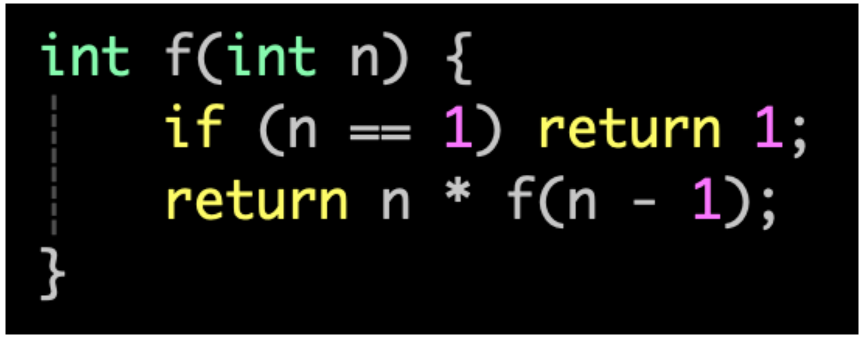


```
int f(int n) {
    if (n == 1) return 1;
    return n * f(n - 1);
}
```

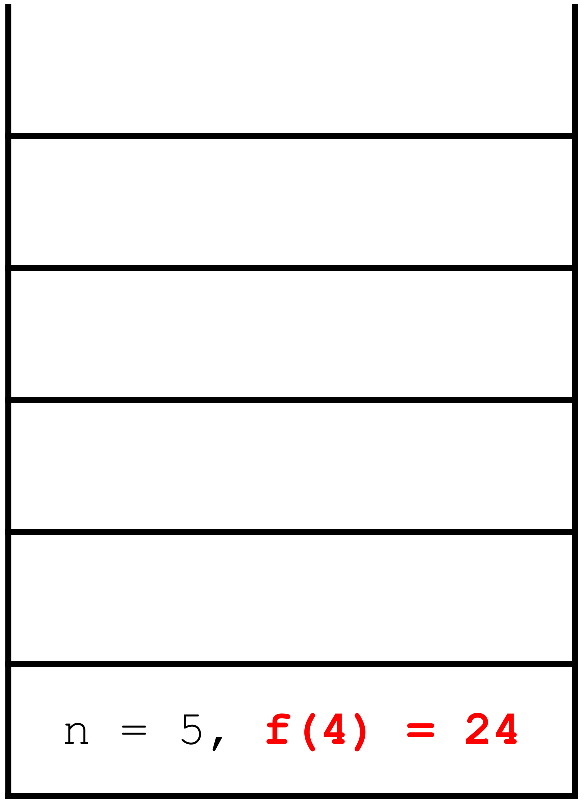


系统栈

用栈模拟递归函数

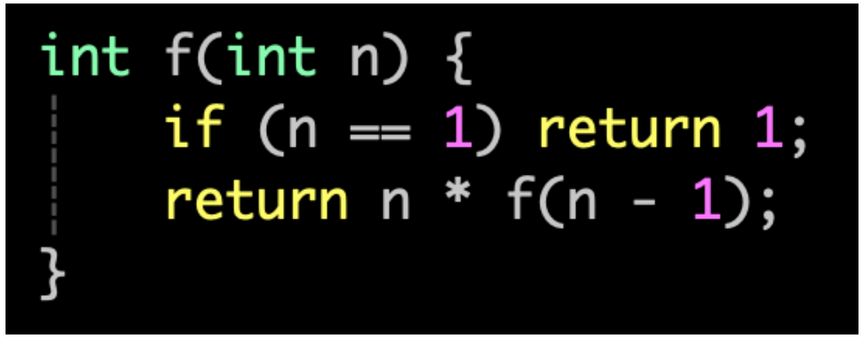


```
int f(int n) {
    if (n == 1) return 1;
    return n * f(n - 1);
}
```

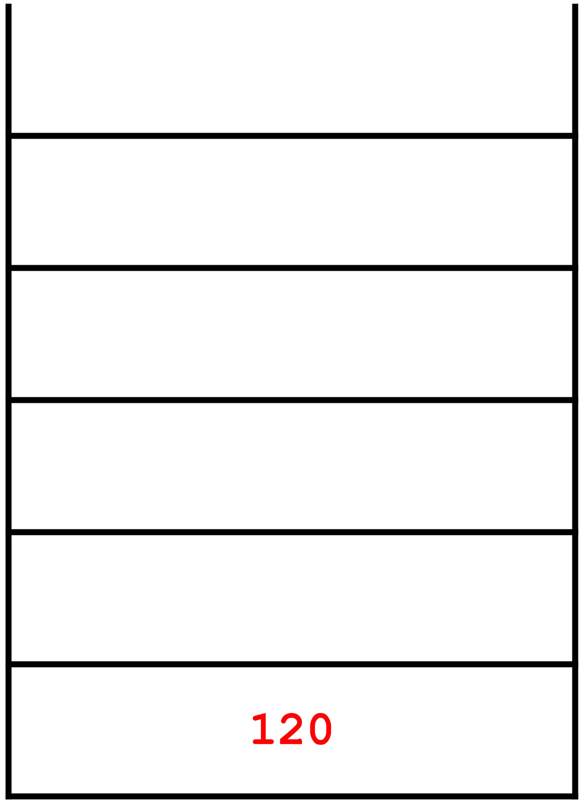


系统栈

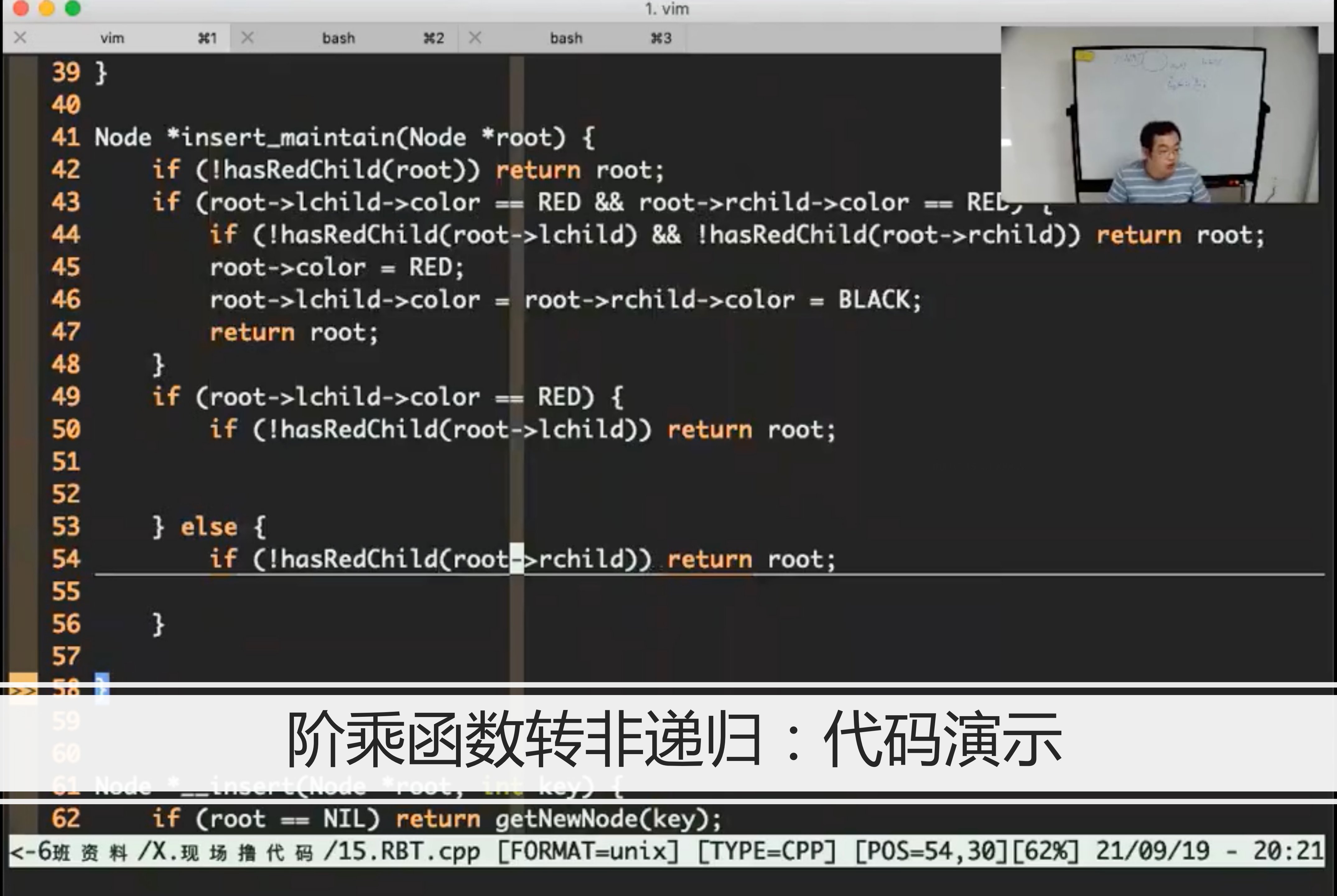
用栈模拟递归函数



```
int f(int n) {
    if (n == 1) return 1;
    return n * f(n - 1);
}
```



系统栈



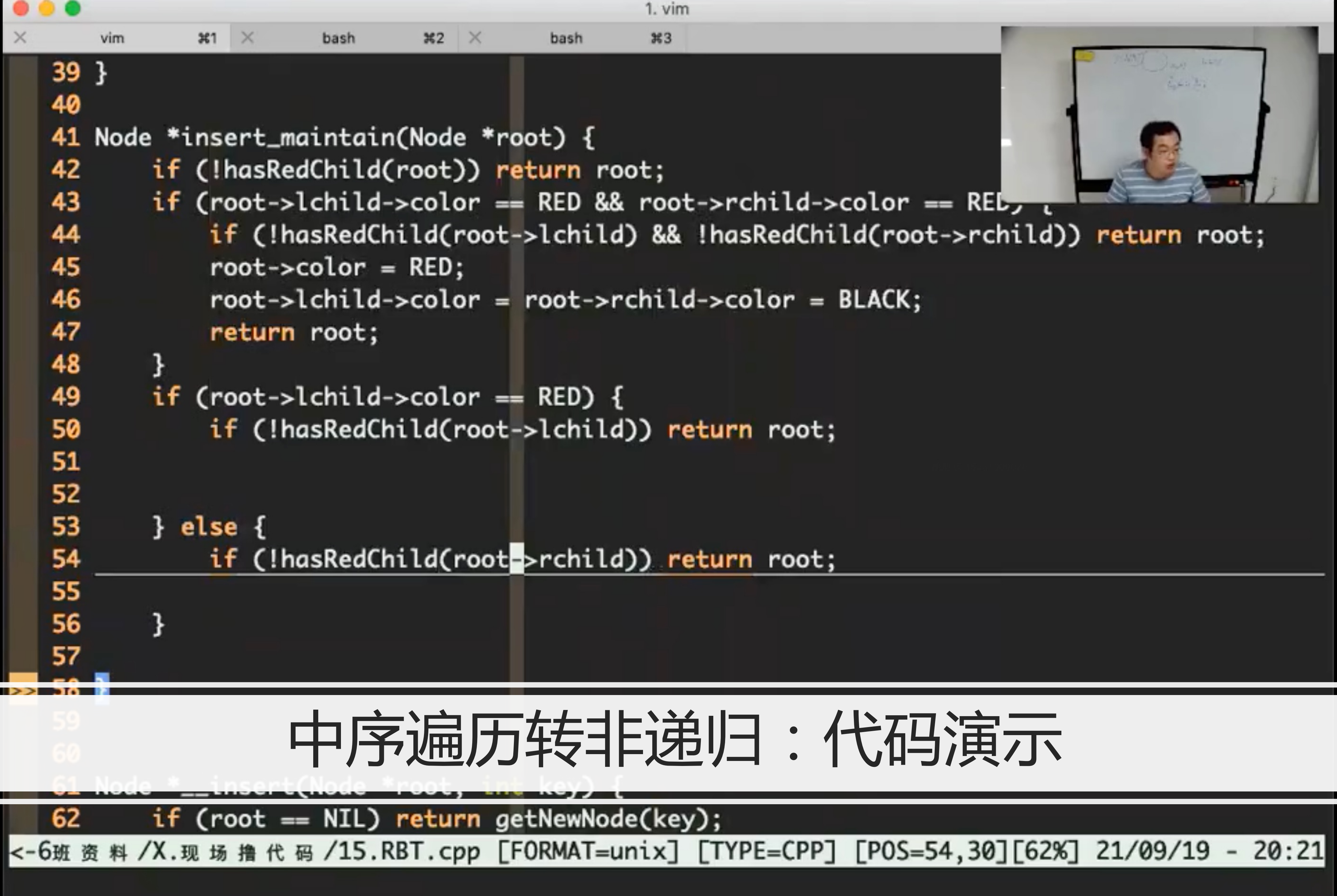
1. vim

vim ⌘1 bash ⌘2 bash ⌘3

```
}

Node *insert_maintain(Node *root) {
    if (!hasRedChild(root)) return root;
    if (root->lchild->color == RED && root->rchild->color == RED) {
        if (!hasRedChild(root->lchild) && !hasRedChild(root->rchild)) return root;
        root->color = RED;
        root->lchild->color = root->rchild->color = BLACK;
        return root;
    }
    if (root->lchild->color == RED) {
        if (!hasRedChild(root->lchild)) return root;


    } else {
        if (!hasRedChild(root->rchild)) return root;

    }

}


Node *__insert(Node *root, int key) {
    if (root == NIL) return getNewNode(key);
```

```
<-6班资料/X.现场撸代码/15.RBT.cpp [FORMAT=unix] [TYPE=CPP] [POS=54,30][62%] 21/09/19 - 20:21
```

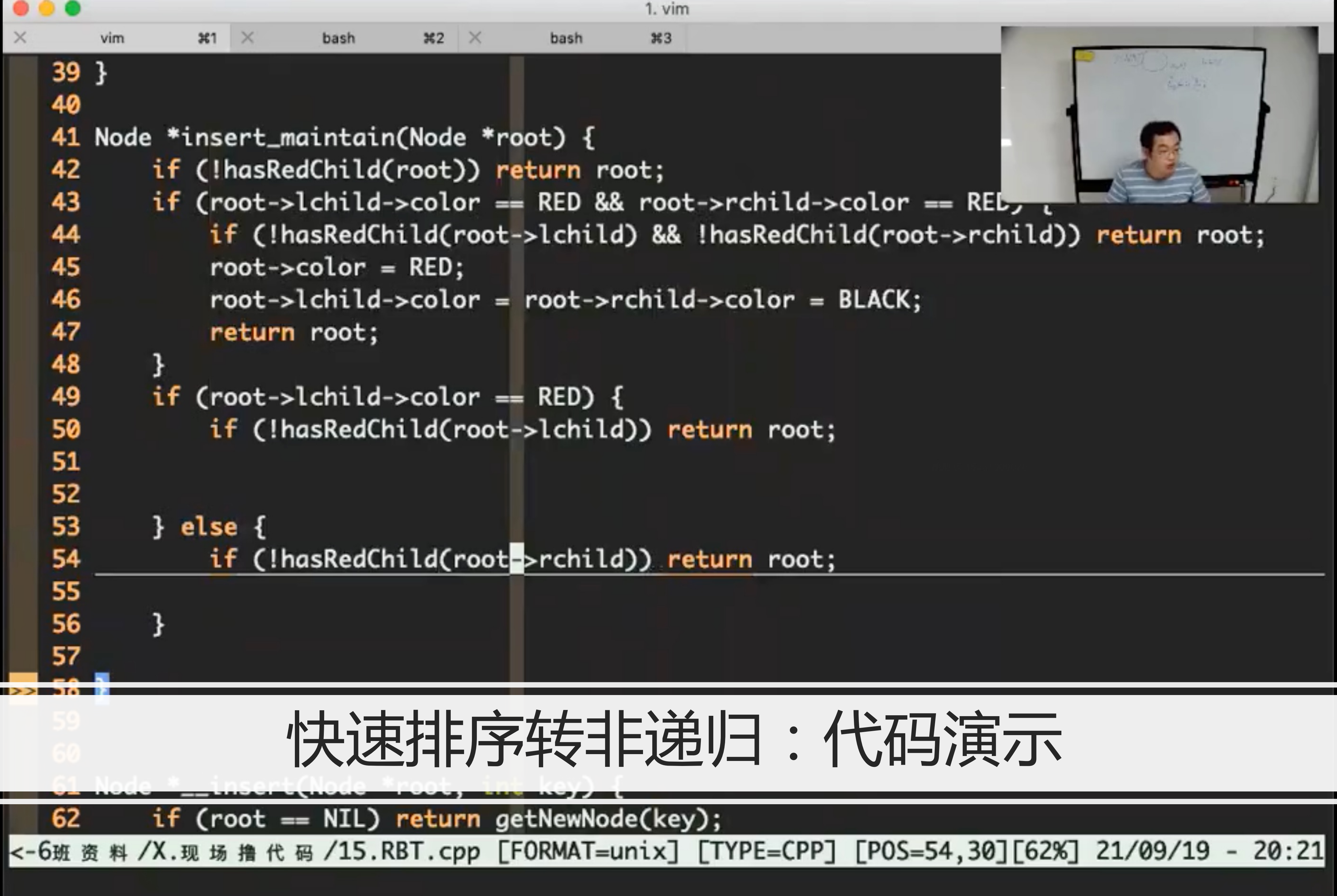
阶乘函数转非递归：代码演示



```
}

Node *insert_maintain(Node *root) {
    if (!hasRedChild(root)) return root;
    if (root->lchild->color == RED && root->rchild->color == RED) {
        if (!hasRedChild(root->lchild) && !hasRedChild(root->rchild)) return root;
        root->color = RED;
        root->lchild->color = root->rchild->color = BLACK;
        return root;
    }
    if (root->lchild->color == RED) {
        if (!hasRedChild(root->lchild)) return root;


    } else {
        if (!hasRedChild(root->rchild)) return root;

    }

}


Node *__insert(Node *root, int key) {
    if (root == NIL) return getNewNode(key);
```

```
<-6班资料/X.现场撸代码/15.RBT.cpp [FORMAT=unix] [TYPE=CPP] [POS=54,30][62%] 21/09/19 - 20:21
```

中序遍历转非递归：代码演示

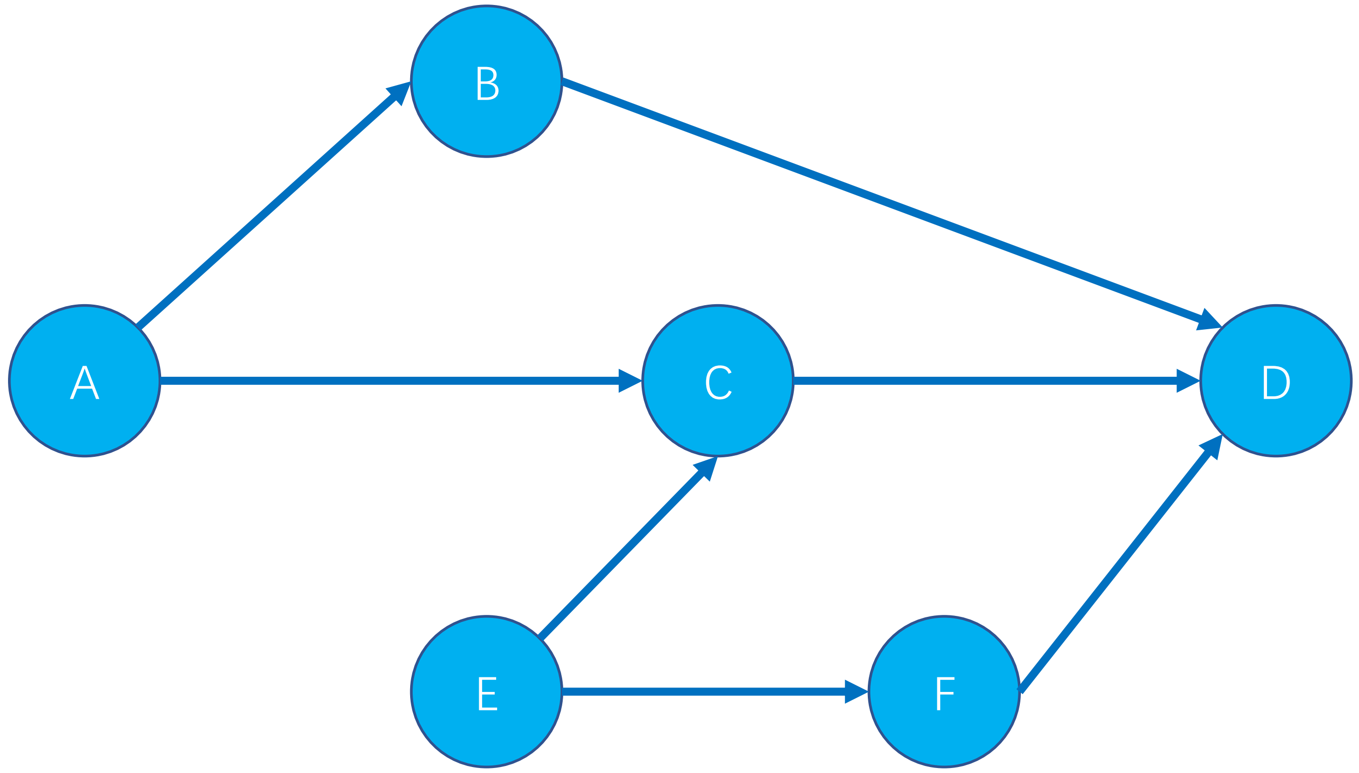


```
}

Node *insert_maintain(Node *root) {
    if (!hasRedChild(root)) return root;
    if (root->lchild->color == RED && root->rchild->color == RED) {
        if (!hasRedChild(root->lchild) && !hasRedChild(root->rchild)) return root;
        root->color = RED;
        root->lchild->color = root->rchild->color = BLACK;
        return root;
    }
    if (root->lchild->color == RED) {
        if (!hasRedChild(root->lchild)) return root;


    } else {
        if (!hasRedChild(root->rchild)) return root;

    }

}


Node *__insert(Node *root, int key) {
    if (root == NIL) return getNewNode(key);
<-6班 资 料 /X.现 场 撸 代 码 /15.RBT.cpp [FORMAT=unix] [TYPE=CPP] [POS=54,30][62%] 21/09/19 - 20:21
```

快速排序转非递归：代码演示

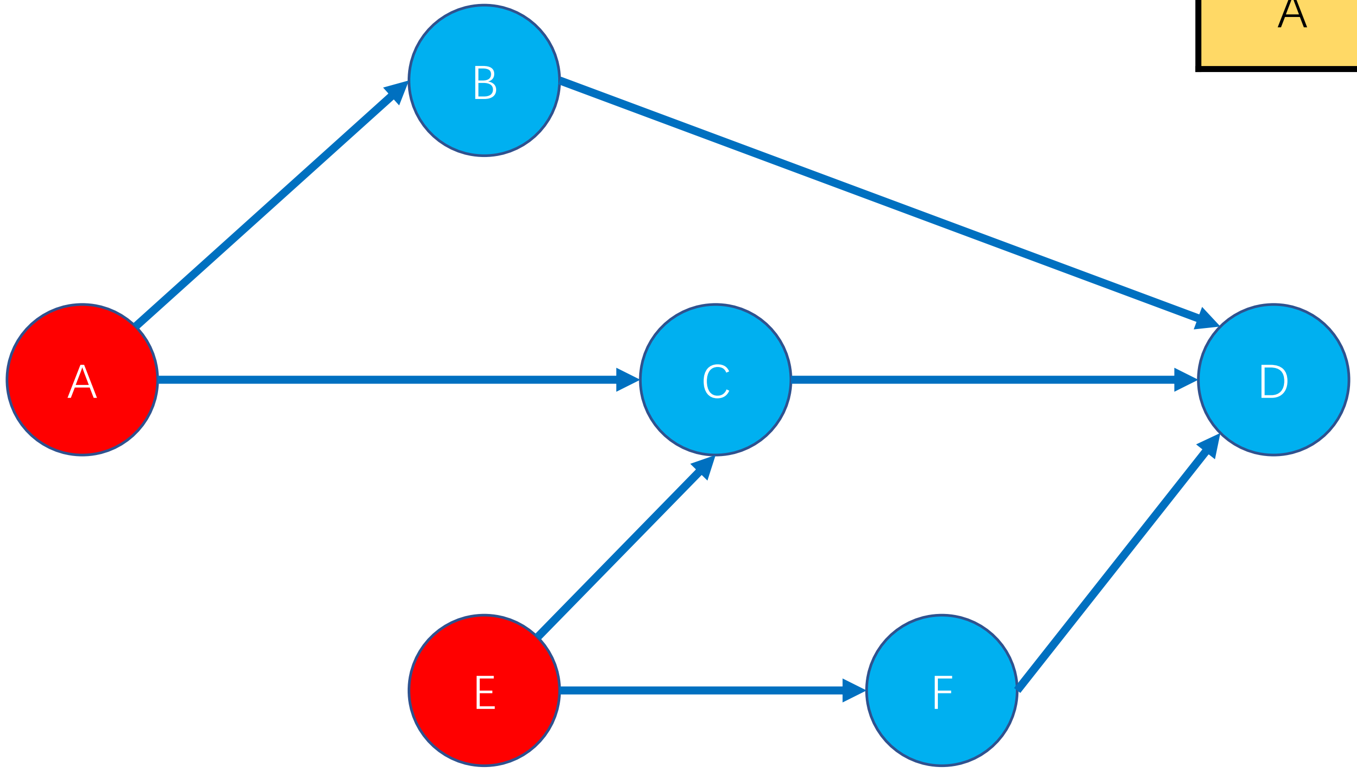
二. 拓扑序分解法

拓扑排序

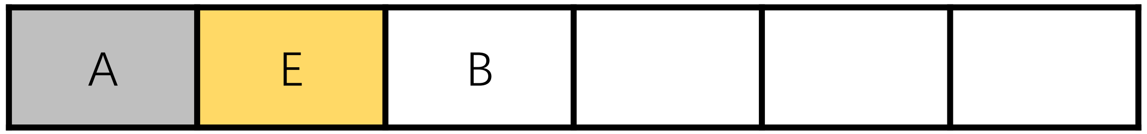


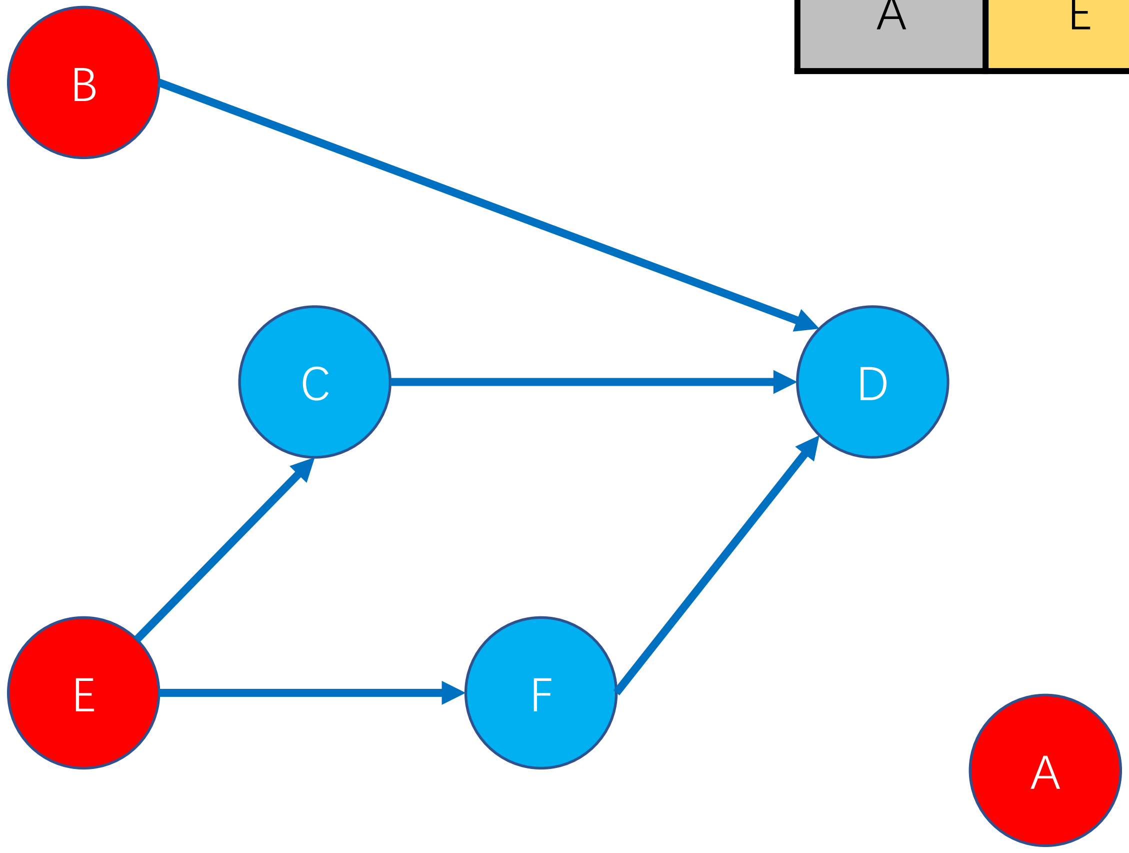
拓扑排序

A	E				



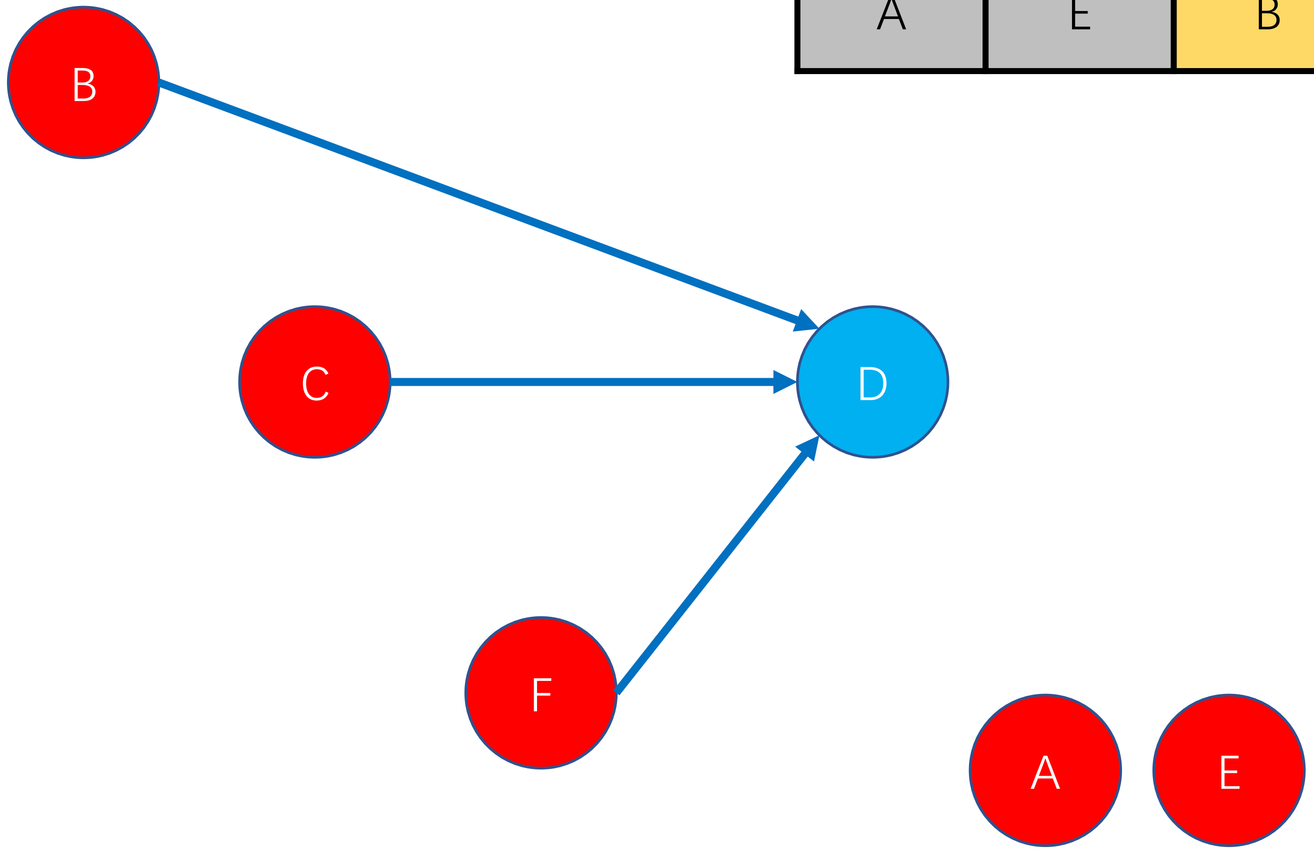
拓扑排序


A	E	B			



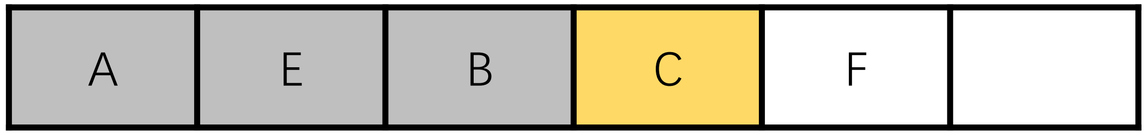
拓扑排序

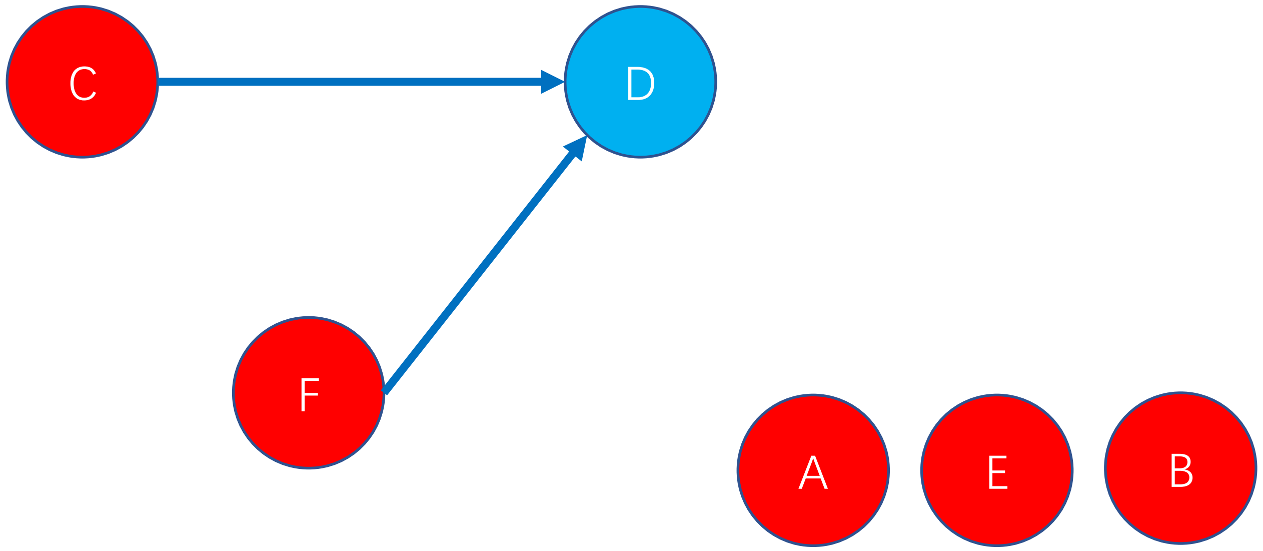
A	E	B	C	F	



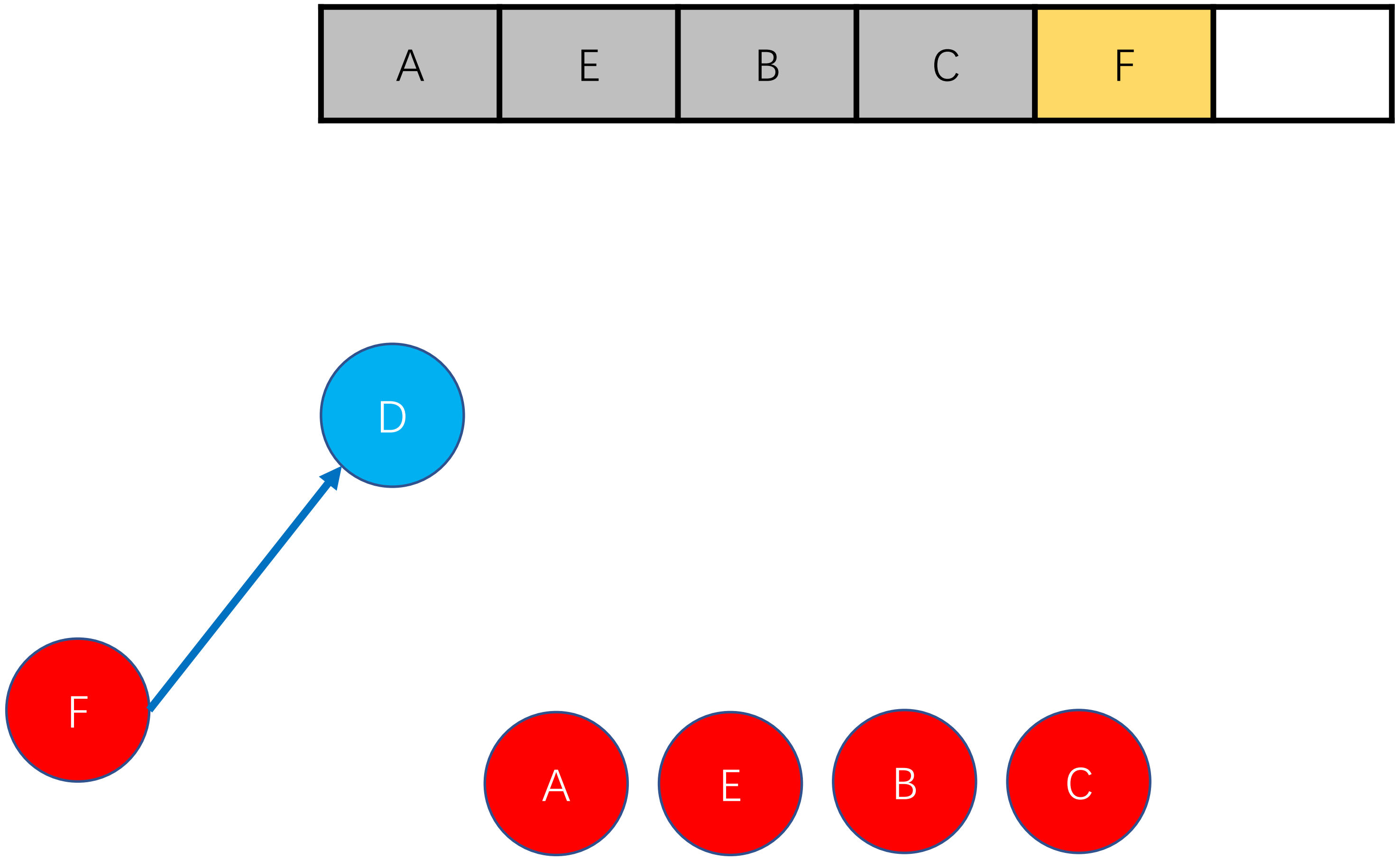
拓扑排序

A	E	B	C	F	

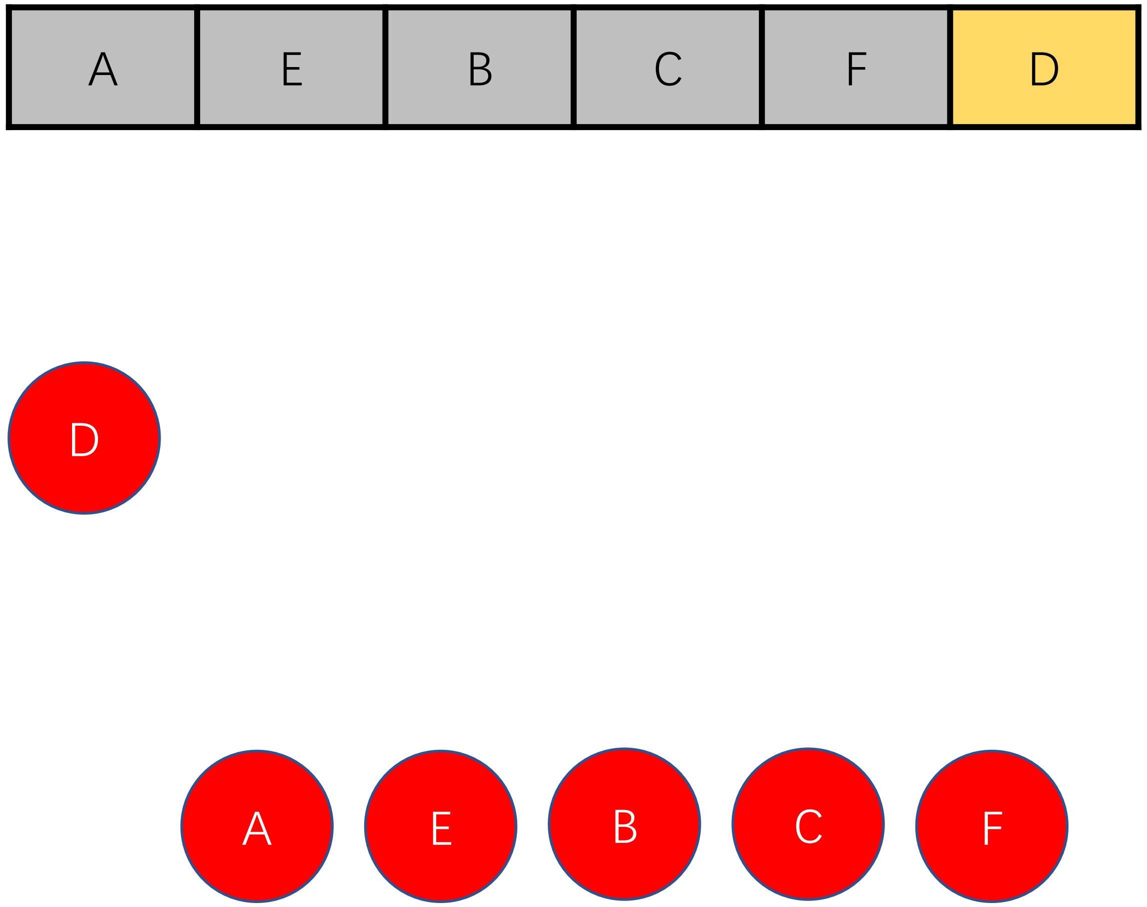




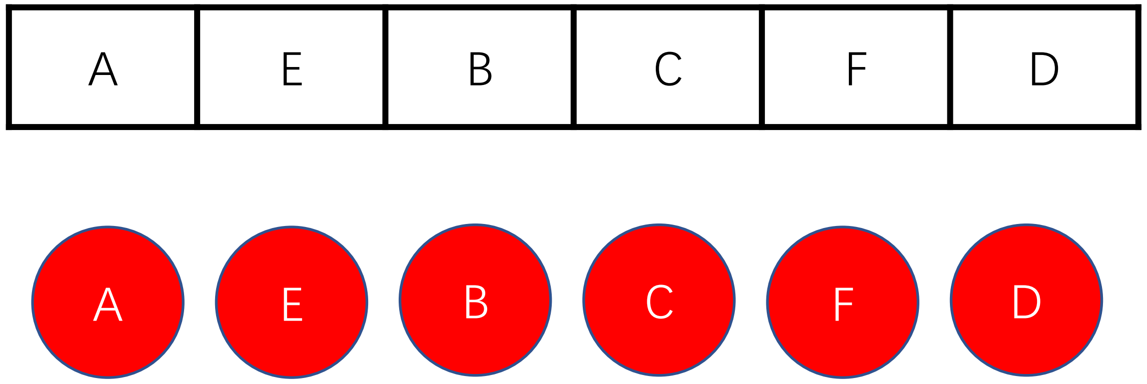
拓扑排序

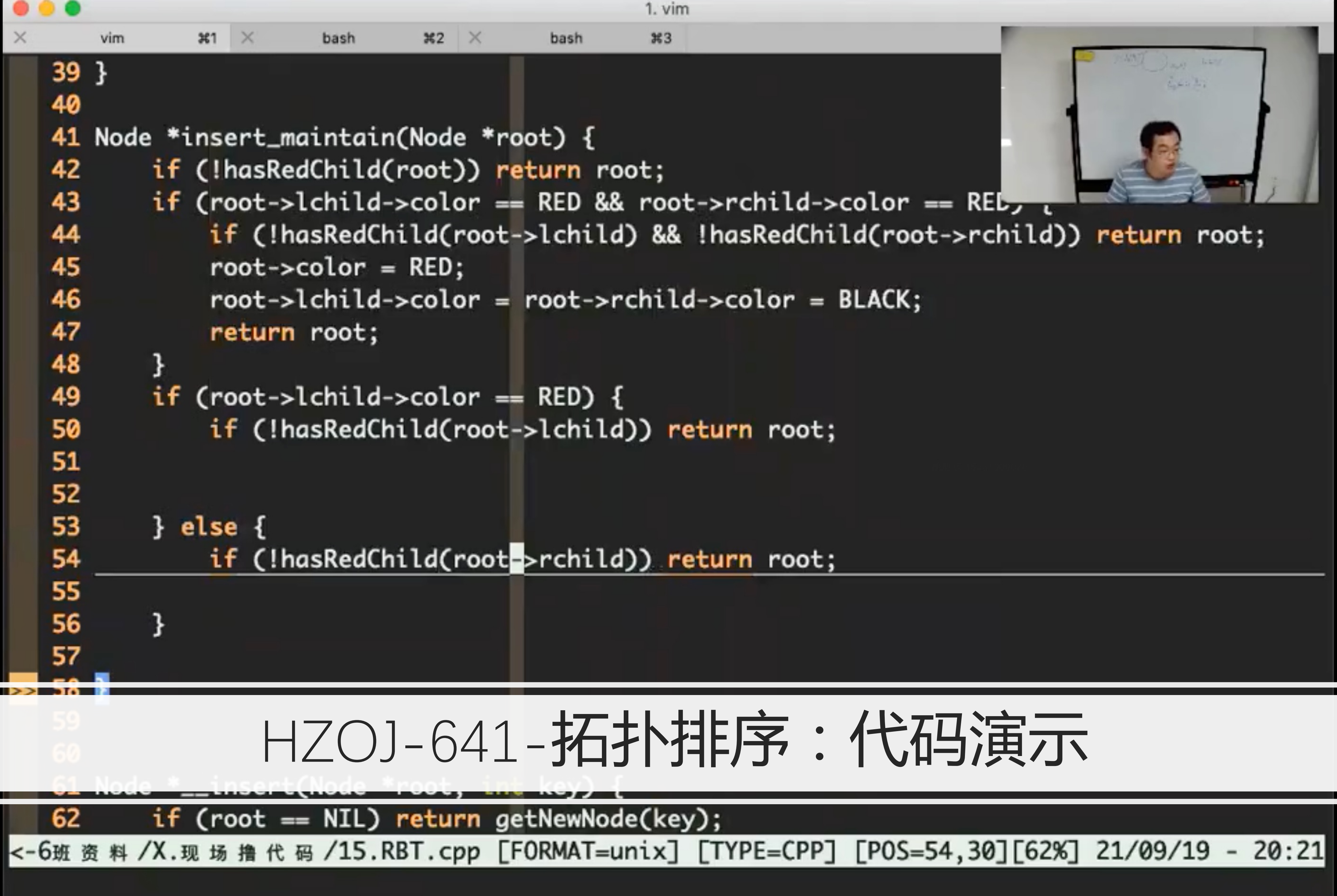


拓扑排序



拓扑排序

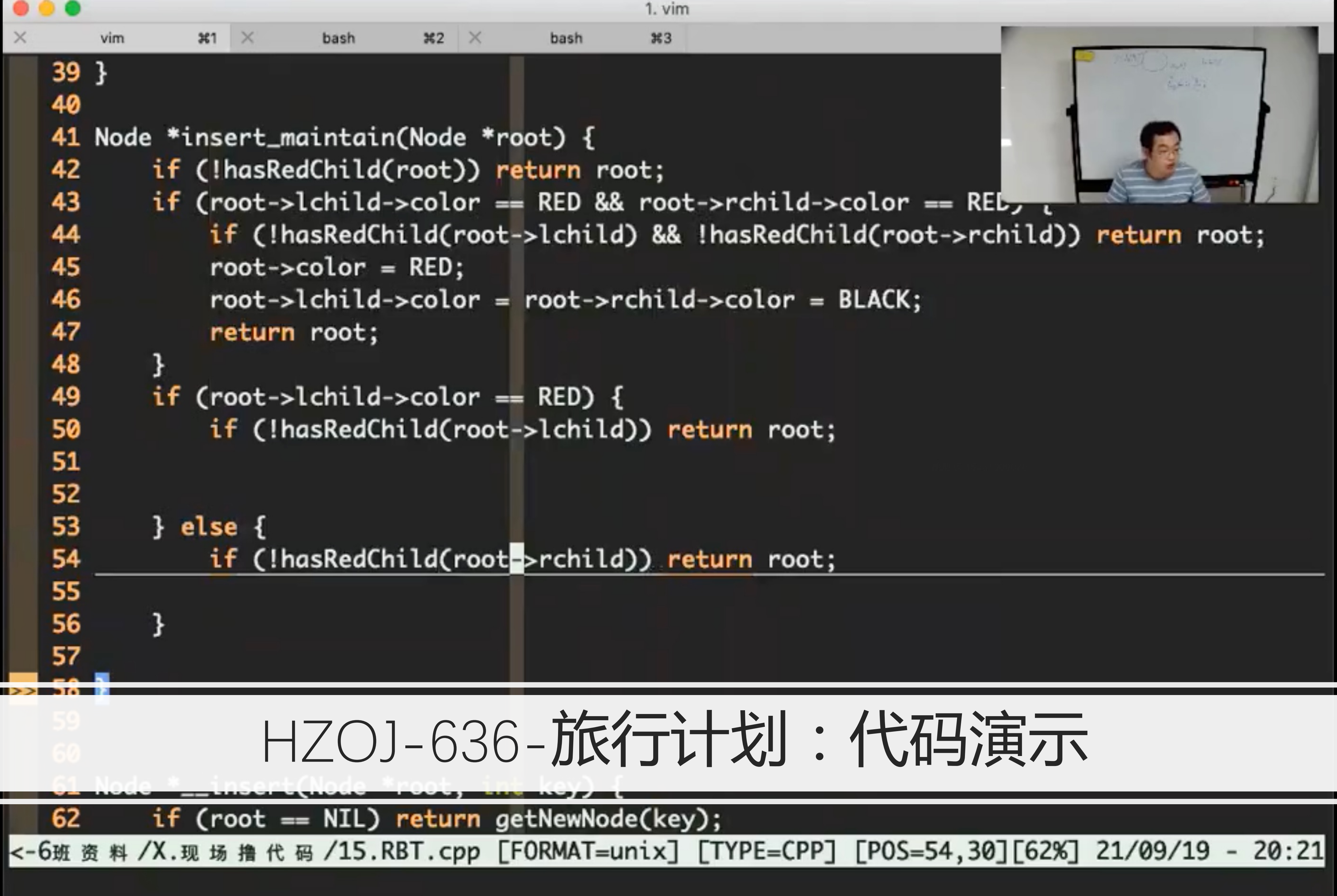




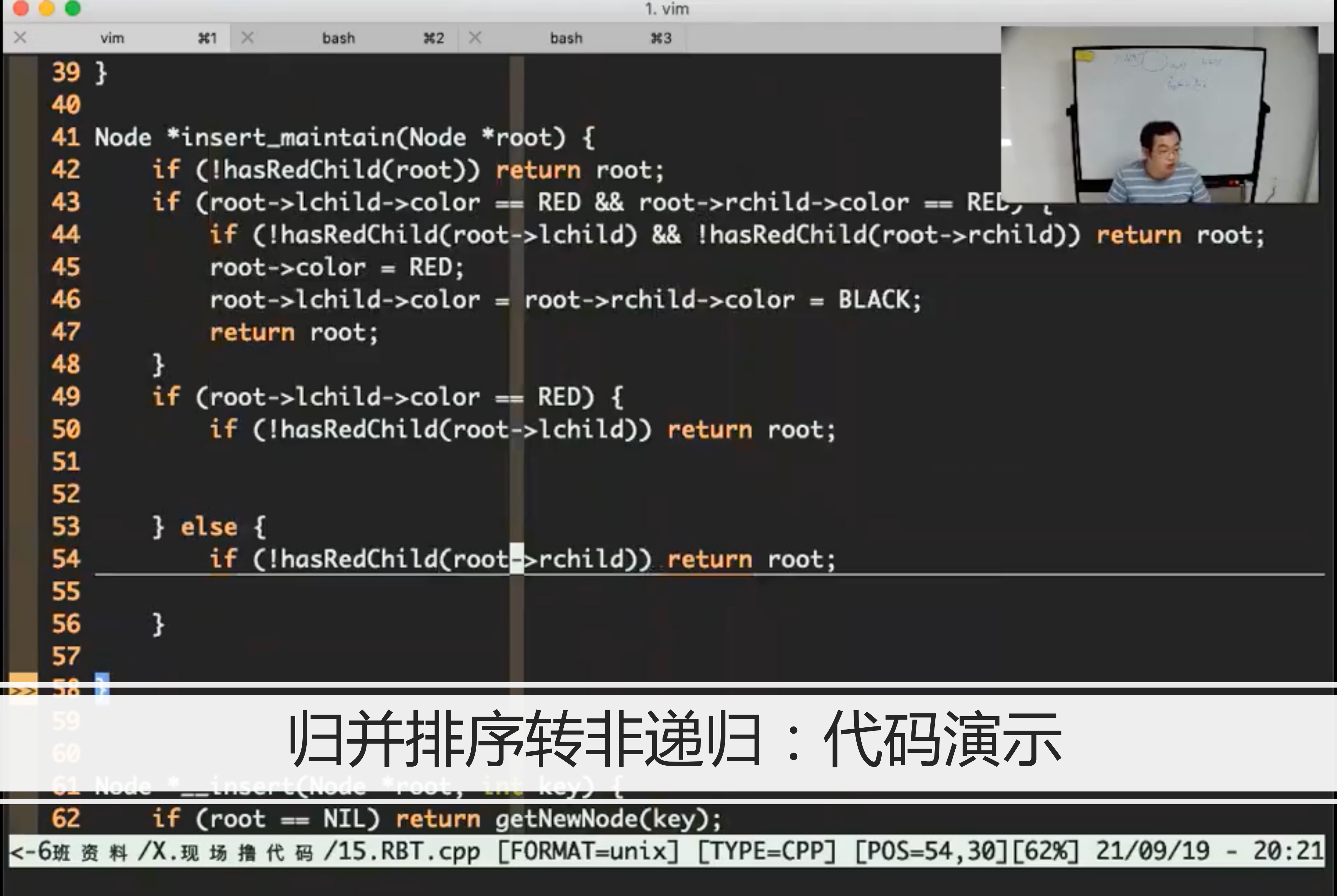
```
}

Node *insert_maintain(Node *root) {
    if (!hasRedChild(root)) return root;
    if (root->lchild->color == RED && root->rchild->color == RED) {
        if (!hasRedChild(root->lchild) && !hasRedChild(root->rchild)) return root;
        root->color = RED;
        root->lchild->color = root->rchild->color = BLACK;
        return root;
    }
    if (root->lchild->color == RED) {
        if (!hasRedChild(root->lchild)) return root;


    } else {
        if (!hasRedChild(root->rchild)) return root;

    }

}


Node *__insert(Node *root, int key) {
    if (root == NIL) return getNewNode(key);
<-6班资料/X.现场撸代码/15.RBT.cpp [FORMAT=unix] [TYPE=CPP] [POS=54,30][62%] 21/09/19 - 20:21
```

HZOJ-641-拓扑排序：代码演示

HZOJ-636-旅行计划：代码演示

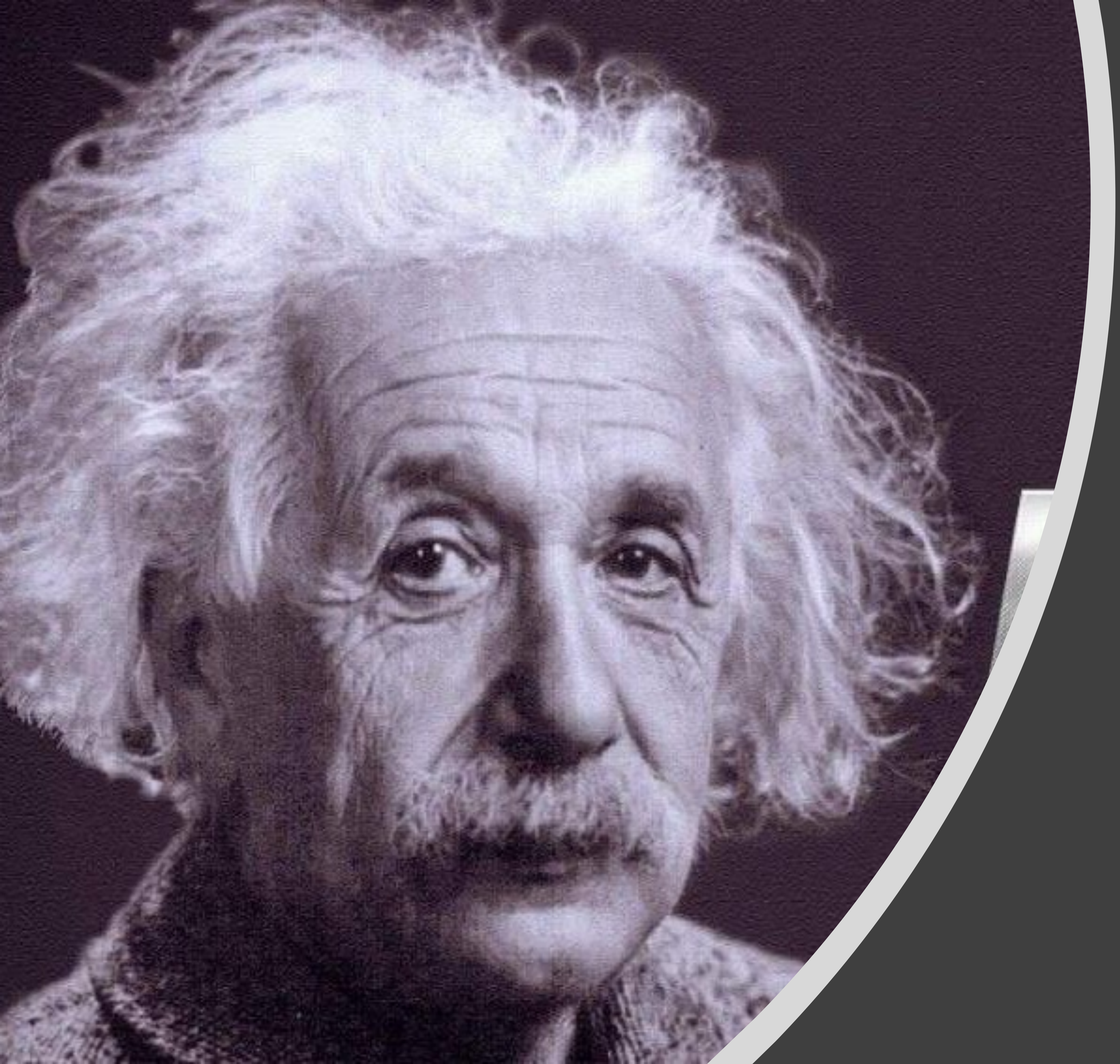


```
}

Node *insert_maintain(Node *root) {
    if (!hasRedChild(root)) return root;
    if (root->lchild->color == RED && root->rchild->color == RED) {
        if (!hasRedChild(root->lchild) && !hasRedChild(root->rchild)) return root;
        root->color = RED;
        root->lchild->color = root->rchild->color = BLACK;
        return root;
    }
    if (root->lchild->color == RED) {
        if (!hasRedChild(root->lchild)) return root;


    } else {
        if (!hasRedChild(root->rchild)) return root;

    }

}


Node *__insert(Node *root, int key) {
    if (root == NIL) return getNewNode(key);
<-6班资料/X.现场撸代码/15.RBT.cpp [FORMAT=unix] [TYPE=CPP] [POS=54,30][62%] 21/09/19 - 20:21
```



```
}

Node *insert_maintain(Node *root) {
    if (!hasRedChild(root)) return root;
    if (root->lchild->color == RED && root->rchild->color == RED) {
        if (!hasRedChild(root->lchild) && !hasRedChild(root->rchild)) return root;
        root->color = RED;
        root->lchild->color = root->rchild->color = BLACK;
        return root;
    }
    if (root->lchild->color == RED) {
        if (!hasRedChild(root->lchild)) return root;


    } else {
        if (!hasRedChild(root->rchild)) return root;

    }

}


Node *__insert(Node *root, int key) {
    if (root == NIL) return getNewNode(key);
<-6班资料/X.现场撸代码/15.RBT.cpp [FORMAT=unix] [TYPE=CPP] [POS=54,30][62%] 21/09/19 - 20:21
```

归并排序转非递归：代码演示



为什么
会出一样的题目？