Computer Science Illuminated
SECOND EDITION
Nell Dale • John Lewis

Chapter 6

Problem Solving and Algorithm Design

（top-down method）

Chapter Goals

- Determine whether a problem is suitable for a computer solution（计算机问题求解）
- Describe the computer problem-solving process（步骤）and relate it to Polya's How to Solve It list
- Distinguish between following an algorithm and developing one
- Apply top-down design methodology（方法） to develop an algorithm to solve a problem in *pseudocode*（*工具*）

6-2

案例研究：遴选班服

6-3

Problem Solving（问题求解）

- **Problem solving** The act of finding a solution to a perplexing(复杂的，令人困惑的), distressing（使痛苦）, vexing（使烦恼）, or unsettled question

6-4

Problem Solving

- G. Polya wrote *How to Solve It: A New Aspect of Mathematical Method*
- His How to Solve It list is quite general
  - Written in the **context** of solving mathematical problems
  - The list becomes **applicable** to all types of problems

6-5

Ask Questions...

- …to understand the problem
  - *What do I know about the problem?*
  - *What is the information that I have to process in order the find the solution?*
  - *What does the solution look like?*
  - *What sort of special cases exist?*
  - *How will I recognize that I have found the solution?*

6-6

案例研究：遴选班服

问题理解：输入、输出是什么？

案例研究：遴选班服

获取信息：？，具体案例：？

学生i可能的投票数据：
给某方案A：1 / 0
给 A，B方案：A > B
给所有方案：C > B > D > A

案例研究：遴选班服

一种直观的解决问题方法的描述：

```
for each 方案x in {A,B,C,D} do
   请大家举手投票；
   count = 0;      //用于统计同意的人数
   for each 学生i in the class do
      if (x[i]==1) then count++;
      if (count 超过半数) then return x;
   end for
end for
return 失败
```

问题求解过程：程序！！

案例研究：遴选班服

现在班上有52个人，程序可能的一些结果：

可能结果1：

方案	count
A	24
B	**38**
C	—
D	—

可能结果2：

方案	count
A	24
B	16
	20
	3

结果验证：测试、分析！！

怎么办……

Look for Familiar Things

- You should **never reinvent the wheel**
- In computing, you see certain problems again and again in different guises
- A good programmer sees a task, or perhaps part of a task (a subtask), that has been solved before and plugs in the solution

相似案例：总统选举

- 方法1：
  1. 提名OBM当总统，大家鼓掌通过；
- 方法2：
  1. 提名OBM，LMN选总统；
  2. 全民投票，票多者胜；
- 方法3：
  1. 提名OBM，LMN，XXX，YYY选总统；
  2. 投你神圣的一票，如果x过半数，x胜出；
  3. 如果没有人过半数，
  4. 留下得票高的2人，返回2；

Divide and Conquer

- Break up a large problem into smaller units that we can handle
  - Applies the concept of abstraction
  - The divide-and-conquer （算法设计基本思想之一，“分治”）approach can be applied over and over again until each subtask is manageable

6-13

案例研究：班服项目

6-14

案例研究：班服项目

6-15

案例研究：班服项目

项目策划
1.收集供应商样品
2.设计主题，色系
3.预期效果
4.确定价格

确定需求
1.各环节负责人
2.大致时间表
3.班服数量
4.预算范围

班服设计
1.设计是艺术，管不了

班服遴选
1.遴选标准
2.程序
1.提交设计
2.统计尺码，数量
3.确定验收标准

班服制作
1.验收组织
2.BUG处理程序

验收确认

细节决定成败！

6-16

案例研究：点菜

- 问题
  - 毕业同学聚餐，大家推举你点菜
- Ask Questions...
- Look for Familiar Things
- Divide and Conquer

6-17

案例研究：“点菜”的一种方案

Ask Questions...

- 我的同学都是哪里人
- 有没有怕辣的，或其他禁忌
- 我们的预算大概是多少
- 餐馆特色，今天的特价菜是什么

Look for Familiar Things

- 餐馆的毕业组合套餐
- 请服务员推荐一些

Divide and Conquer

1、广东菜
1.1 茶、汤
1.2 鸡、肉、鱼、虾、嗜嗜、下饭菜
1.3 小点、凉菜
2、四川特色菜
2.1 辣子鸡，麻婆豆腐
3、时蔬
4、主食与饮料
4.1 面点
4.2 米饭
4.3 饮料

6-18

Algorithms（算法）

- **Algorithm** A set of instructions for solving a problem or subproblem in a finite amount of time using a finite amount of data
- The instructions must be unambiguous

6-19

案例研究：遴选班服

一种直观的解决问题过程的描述：

```
for each 方案x in {A,B,C,D} do
   请大家举手投票；
   count = 0;      //用于统计同意的人数
   for each 学生i in the class do
      if (x[i]==1) then count++;
      if (count 超过半数) then return x；
   end for
end for
return 失败
```

6-20

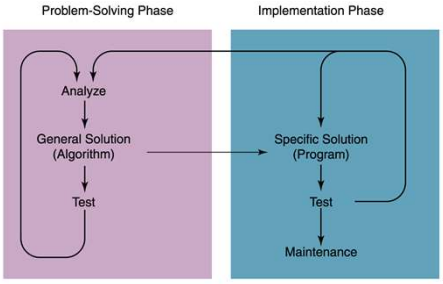
Computer Problem-Solving

Algorithm Development Phase		
	Analyze	Understand (define) the problem.
	Propose algorithm	Develop a logical sequence of steps to be used to solve the problem.
	Test algorithm	Follow the steps as outlined to see if the solution truly solves the problem.
Implementation Phase		
	Code	Translate the algorithm (the general solution) into a programming language.
	Test	Have the computer follow the instructions. Check the results and make corrections until the answers are correct.
Maintenance Phase		
	Use	Use the program.
	Maintain	Modify the program to meet chaining requirements or to correct any errors.

Figure 6.2 **The computer problem-solving process**

6-21

Figure 6.3: The Interactions Between Problem-Solving Phases



6-22

Pseudocode（伪代码）

- Uses a mixture of English and formatting to make the steps in the solution explicit

```
While (the quotient is not zero)
    Divide the decimal number by the new base
    Make the remainder the next digit to the left in the answer
    Replace the original decimal number with the quotient
```

6-23

案例：算法设计

- 问题描述
  - 计算 1+2+……+n 的和
- 数学与计算分解
  - (…((((0+1)+2)+3)+4)+…+n)
- Algorithm
  - *set* sum to 0
  - *for* count *from* 1 *to* n
    - *set* sum to sum + count
  - *end for*
  - *output* sum

6-24

伪代码与编程

- 视频

https://www.khanacademy.org/cs/programming/good-practices/p/pseudo-code

6-25

Developing an Algorithm

- The plan must be suitable in a suitable form
- Two methodologies that currently used
  - Top-down design
  - Object-oriented design

6-26

Top-Down Design

- Breaking the problem into a set of subproblems called **modules**
- Creating a hierarchical structure of problems and subproblems (modules)

6-27

Top-Down Design

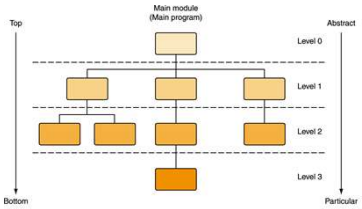


Figure 6.5 An example of top-down design

- This process continues for as many levels as it takes to expand every task to the smallest details
- A step that needs to be expanded is an abstract step

6-28

A General Example

- Planning a large party

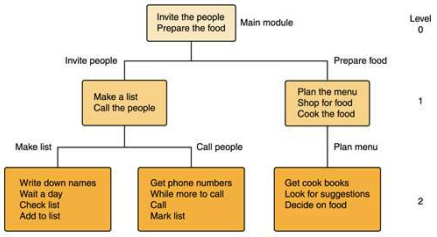


Figure 6.6 Subdividing the party planning

6-29

Testing the Algorithm

- The process itself must be tested
- Testing at the algorithm development phase involves looking at each level of the top-down design

6-30

Testing the Algorithm

- **Desk checking** Working through a design at a desk with a pencil and paper
- **Walk-through** Manual simulation of the design by the team members, taking sample data values and simulating the design using the sample data
- **Inspection** One person (not the designer) reads the design (handed out in advance) line by line while the others point out errors

6-31

作业 1/2

1、阅读 Pseudocode Standard。（答案可以打印）
1）用伪代码描述将十进制转换成16进制的方法
2）C语言实现（先用注释写好算法，然后翻译）
3）使用 -1，0，1，15，26，3265 最为输入测试你的程序

2、名词解释与对比
1）Top-down design
2） Work breakdown structure (WBS)
3）简述管理学 WBS 与 信息学Top-down设计 的异同

3、仔细观察您洗衣机的运作过程，运用Top-down设计方法和Pseudocode描述洗衣机控制程序。假设洗衣机可执行的基本操作如下：
water_in_switch(open_close) // open 打开上水开关，close关闭
water_out_switch(open_close) // open 打开排水开关，close关闭
get_water_volume() //返回洗衣机内部水的高度

6-32

作业 2/2

motor_run(direction) // 电机转动。left左转，right右转，stop停
time_counter() // 返回当前时间计数，以秒为单位
halt(returncode) //停机，success 成功 failure 失败

1）请使用伪代码分解“正常洗衣”程序的大步骤。包括注水、浸泡等
2）进一步用基本操作、控制语句（IF、FOR、WHILE等）、变量与表达式，写出每个步骤的伪代码
3）根据你的实践，请分析“正常洗衣”与“快速洗衣”在用户目标和程序上的异同。你认为是否存在改进（创新）空间，简单说明你的改进意见？
4）通过步骤3），提取一些共性功能模块（函数），简化“正常洗衣”程序，使程序变得更利于人类理解和修改维护。例如：
wait(time) //等待指定的时间；
注水(volume,timeout) //在指定时间内完成注水，否则停机；
排水(timeout)。 等子程序

预习：下一节课Object-Oriented Design。在project1中找object, class, field(property) ,method 等概念的具体实例(instance)。

6-33