01 Course Overview

College of Computer Science, CQU

Outline

- About teacher
- Textbook & Reference materials
- Grading policy
- Course introduction---problem-based

Welcome to the course

- Jia Lee(李 佳)
- Email: lijia@cqu.edu.cn
- Phone: 13883315369

课程交流、随堂小测试平台：

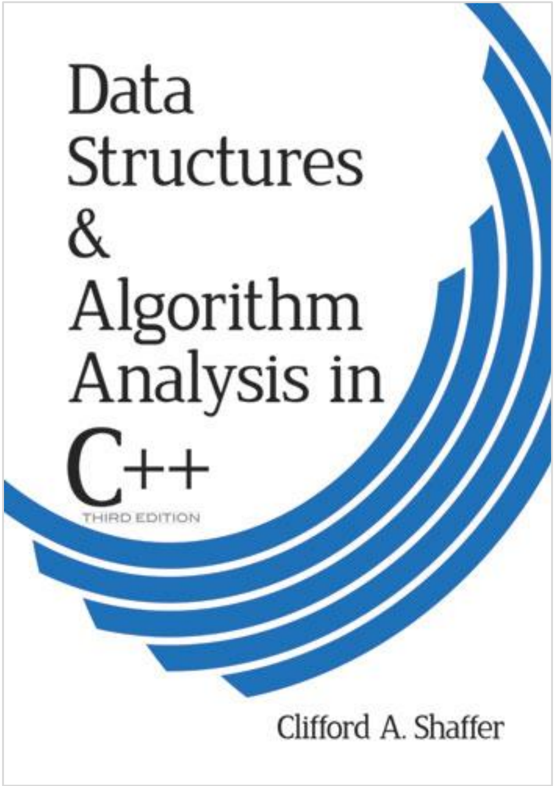
- 云班课：4794155
- QQ群：数据结构课程群

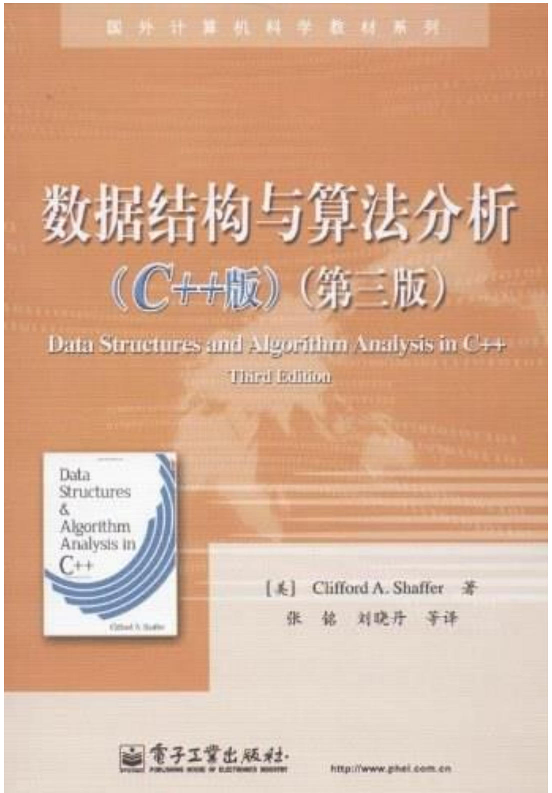
在线编程平台：

- CQUOJ: acm.cqu.edu.cn
- PTA: pintia.cn

The Book

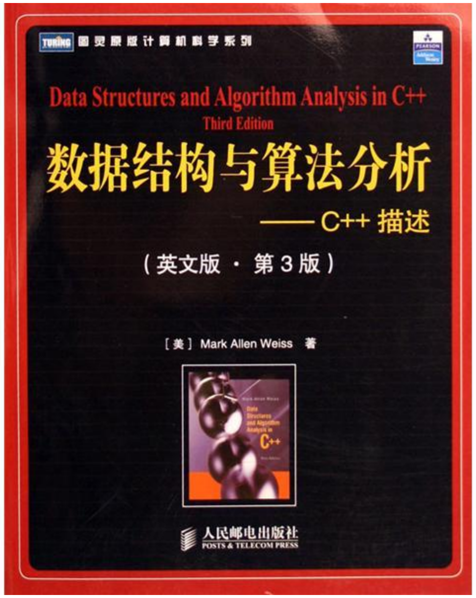
- Data Structures and Algorithm Analysis (C++ Version)
- Clifford A. Shaffer

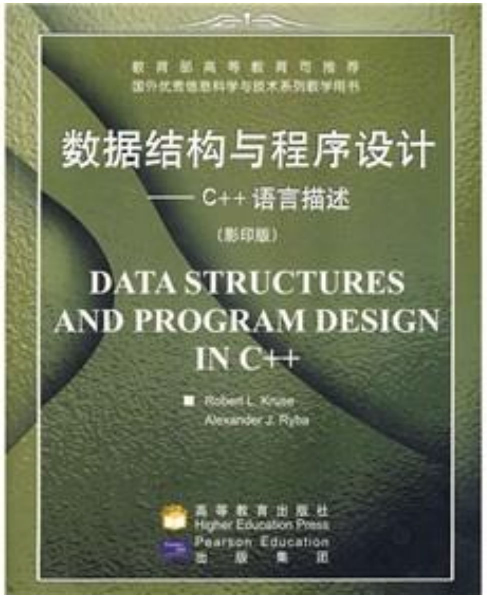


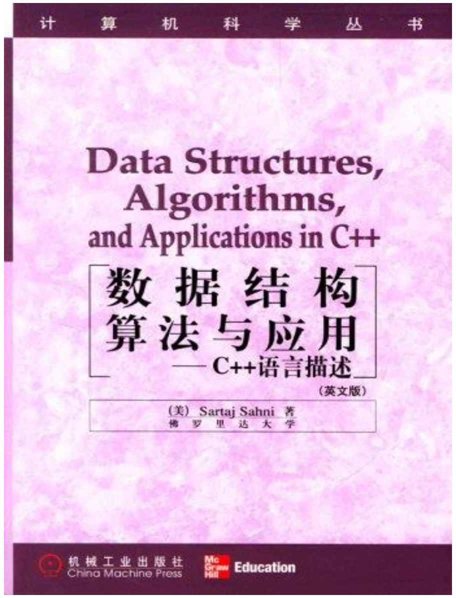


Reference Books

- Data Structure and Algorithm Analysis in C++ (Third Edition)，Mark Allen Weiss， Pearson Education, 2006.
- Data Structures, Algorithms, and Applications in C++，Sartaj Sahni，McGraw-Hill, 1998.
- 《数据结构（ C 语言版）》，严蔚敏，吴伟民编著，清华大学出版社，2007年第1版







Grading policy

- Final Examination 55%

  主要考察分析问题、建模、应用数据结构的计算思维能力

- Projects 25%

  考察较大规模问题的分析与表达能力，应用多种数据结构的综合能力

- Homeworks+Quiz 20%

  每周布置一次课后在线编程，4-6题（含1-2道竞赛题或面试题）

  考察选择或设计数据结构和算法解决复杂问题的能力

  随堂小测试

  及时考察对课堂知识点的理解和掌握情况

1.1Course Description

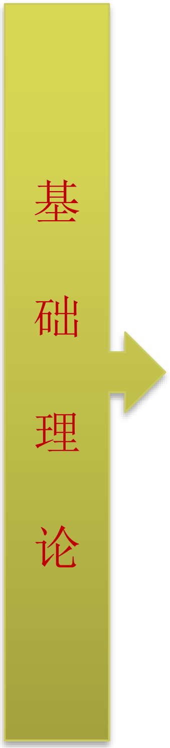
- This course is emphasis on how to design, implementation and run-time analysis of important data structures and algorithms.

- The main data structures considered include lists, stacks, queues, trees and graphs.

Course Contents

- □ Algorithm, Time Complexity

  (算法与时间复杂度)
- □ Linear List(线性表)
- □ Stack and Queue(栈与队列)
- □ Tree (树)
- □ Graph(图)
- □ Sort (排序)
- □ Search（查找）



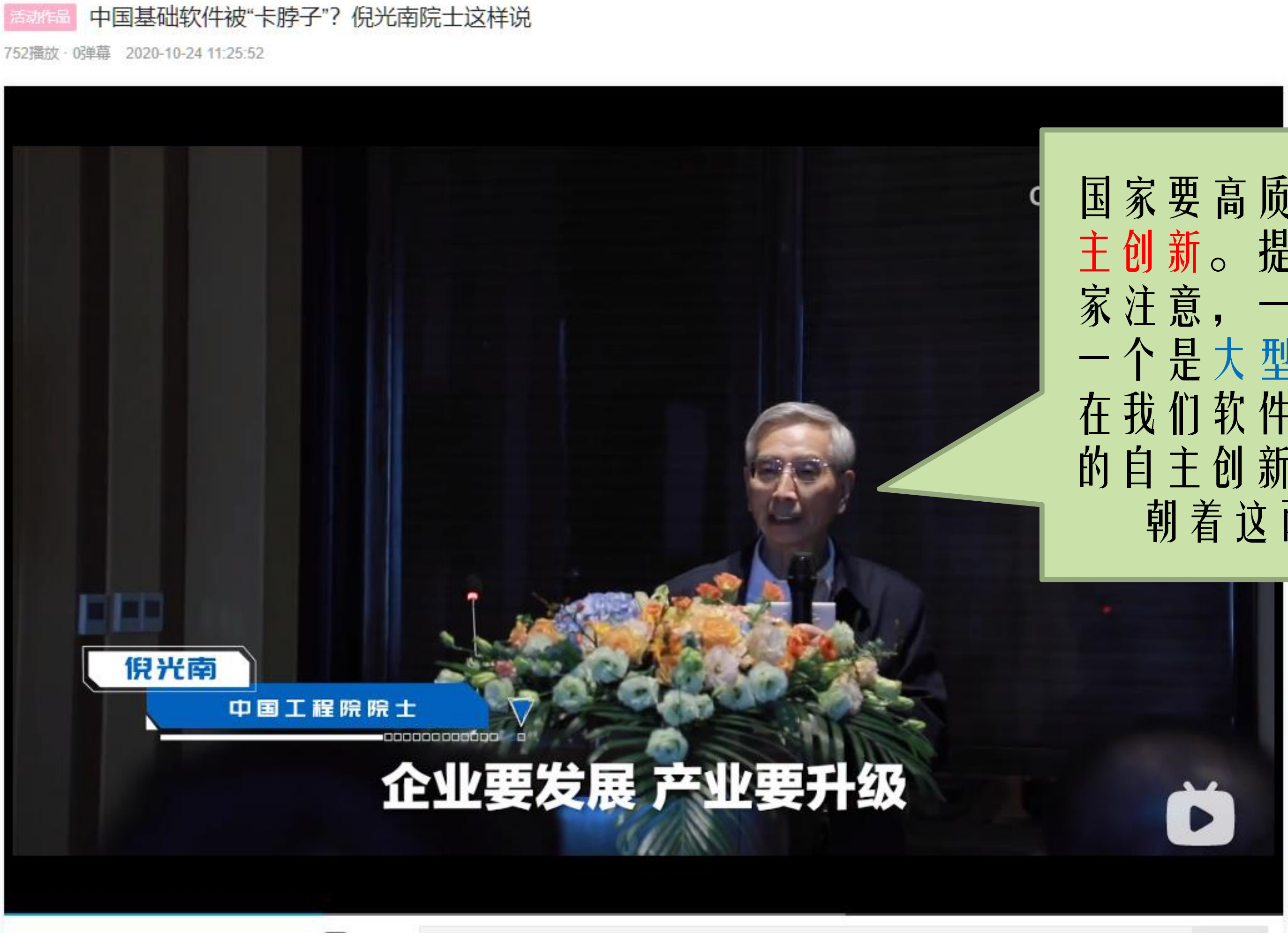
基础理论

- 基础软件

  操作系统、数据库、办公软件等
- 工业软件

  ERP, EDA, Alias, etc

Ultimate Goal



活动作品 中国基础软件被“卡脖子”？倪光南院士这样说

752播放 · 0弹幕 2020-10-24 11:25:52

国家要高质量发展，要靠自主创新。提取两个关键词大家注意，一个是中国原创，一个是大型基础软件。今后在我们软件领域，更高水平的自主创新、自力更生，要朝着这两方面努力。

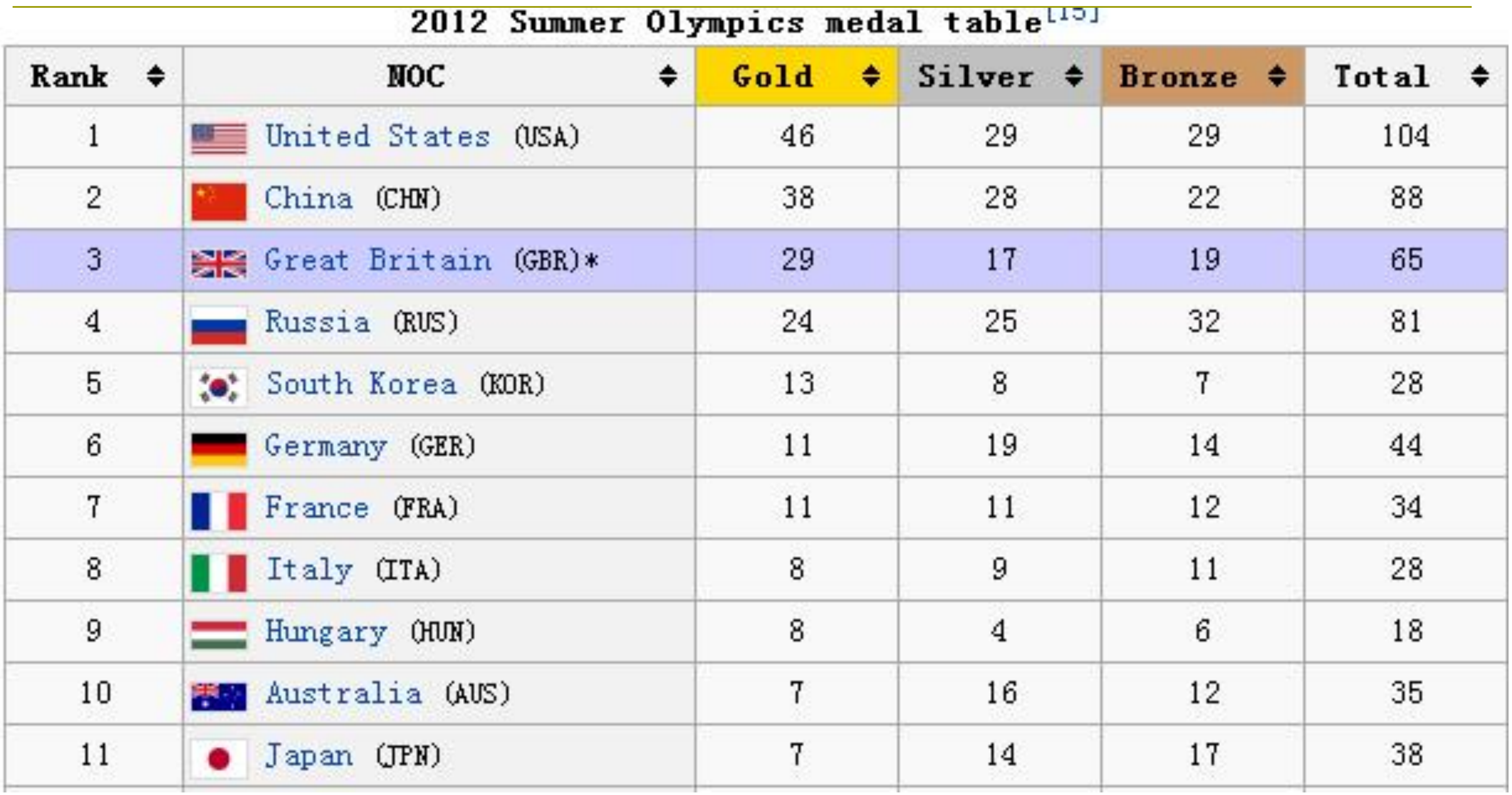
Goals

- To present the commonly used data structures.
  - These form a programmer's basic data structure "toolkit." For many problems, some data structure in the toolkit provides a good solution.

- To introduce the idea of tradeoffs and reinforce the concept that there are costs and benefits associated with every data structure.
  - This is done by describing the amount of space and time required for typical operations.

- To teach how to measure the effectiveness of a data structure or algorithm.
  - Only through such measurement can you determine which data structure in your toolkit is most appropriate for a new problem.

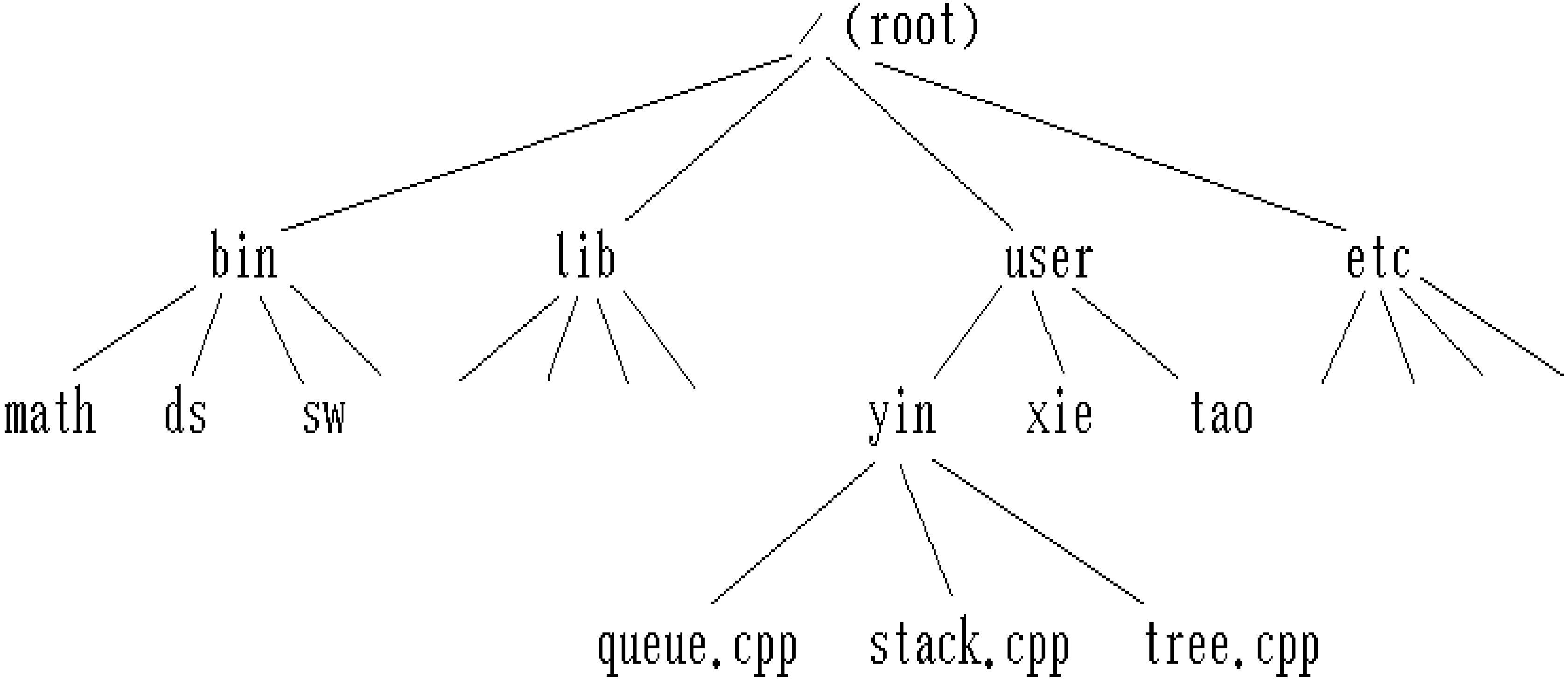
Application Example



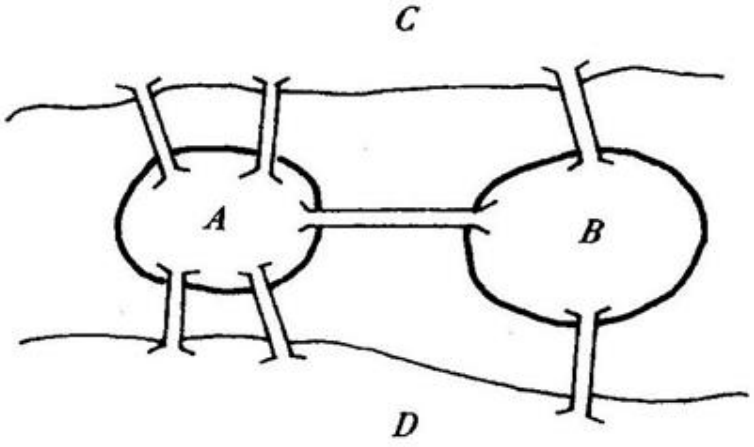
2012 Summer Olympics medal table[15]

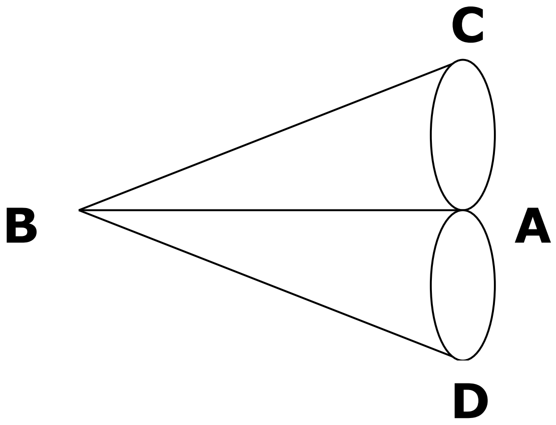
Rank	NOC	Gold	Silver	Bronze	Total
1	United States (USA)	46	29	29	104
2	China (CHN)	38	28	22	88
3	Great Britain (GBR)*	29	17	19	65
4	Russia (RUS)	24	25	32	81
5	South Korea (KOR)	13	8	7	28
6	Germany (GER)	11	19	14	44
7	France (FRA)	11	11	12	34
8	Italy (ITA)	8	9	11	28
9	Hungary (HUN)	8	4	6	18
10	Australia (AUS)	7	16	12	35
11	Japan (JPN)	7	14	17	38

Application Example



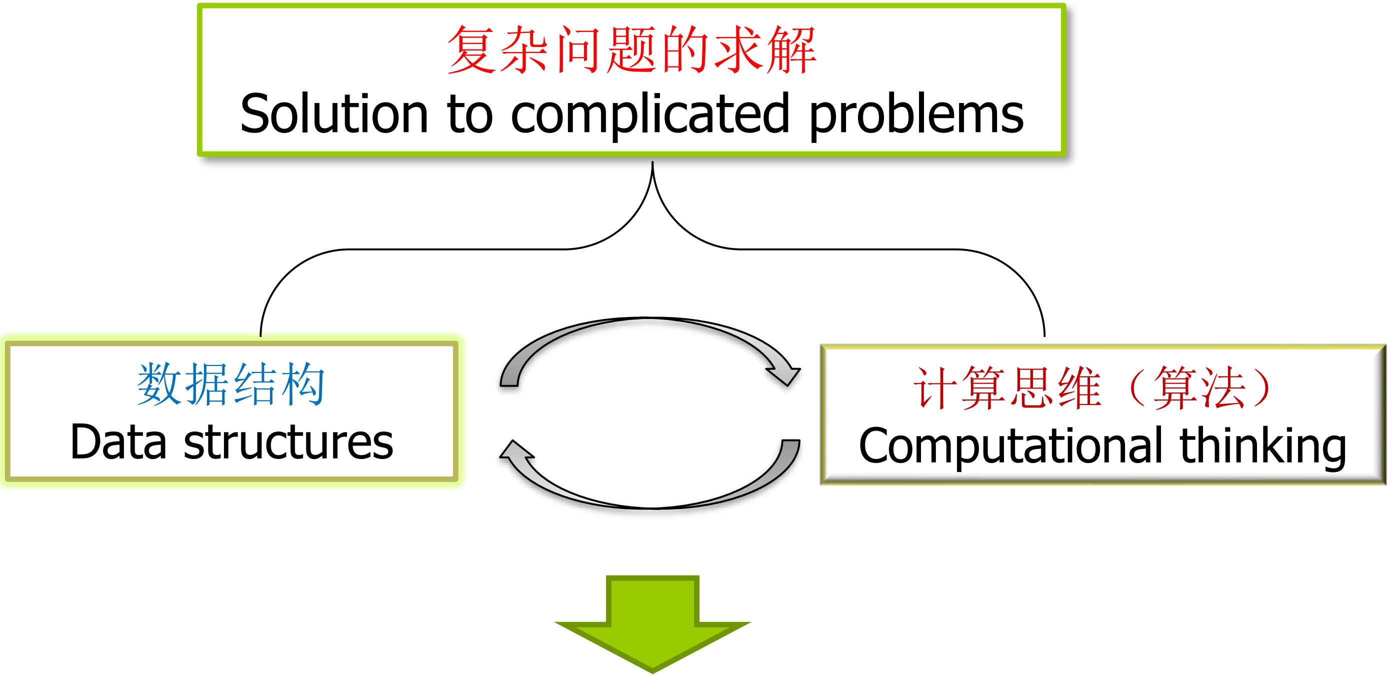
Application Example





Seven Bridges Problem

The Need for Data Structures



复杂问题的求解

Solution to complicated problems

数据结构

Data structures

计算思维（算法）

Computational thinking

- 没有好的数据结构，不能激发出好的计算思维（算法）
- 没有好的计算思维，不会正确使用工具

The Need for Data Structures

- An intriguing example!

质数问题是常见的面试题和竞赛题！！

有二位质数，甲只知道其十位数，乙只知道其个位数。通过下面的甲和乙的对话，推断该二位质数是什么？

甲：我猜不出来。

乙：我早就知道你猜不出来！

甲：我还是猜不出来。。。

乙：我早就知道你还是猜不出来！

甲：喔，我猜出来了！

乙：我也知道了。。

问：是什么质数？

The Need for Data Structures

有二位质数，甲只知道十位数，乙只知道个位数。关于该质数，甲和乙有下述对话：

甲：我猜不出来。

乙：我早就知道你猜不出来！

甲：我还是猜不出来。。。

乙：我早就知道你还是猜不出来！

甲：喔，我猜出来了！

乙：我也知道了。。

最简单的数据结构：

用一维数组枚举所有二位质数

11、13、17、19、23、29、31、37、41、43、47、53、59、61、67、71、73、79、83、89、97

问题点：　很难发现规律

The Need for Data Structures

有二位质数，甲只知道十位数，乙只知道个位数。关于该质数，甲和乙有下述对话：

Q: 个位只需要考虑奇数，且5除外？

甲：我猜不出来。

乙：我早就知道你猜不出来！

甲：我还是猜不出来。。。

乙：我早就知道你还是猜不出来！

甲：喔，我查出来了！

乙：我也知道了。。

十 个	1	3	7	9
1	11	13	17	19
2		23		29
3	31		37	
4	41	43	47	
5		53		59
6	61		67	
7	71	73		79
8		83		89
9			97	

把十位与个位分开，用二维表格列举所有质数

The Need for Data Structures

有二位质数ab，甲只知道十位数a，乙只知道个位数b。关于该质数，甲和乙有下述对话：

甲：我猜不出来。

乙：我早就知道你猜不出来！

甲：我还是猜不出来。。。

乙：我早就知道你还是猜不出来！

甲：喔，我猜出来了！

乙：我也知道了。。

十 个	1	3	7	9
1	11	13	17	19
2		23		29
3	31		37	
4	41	43	47	
5		53		59
6	61		67	
7	71	73		79
8		83		89
9			97	

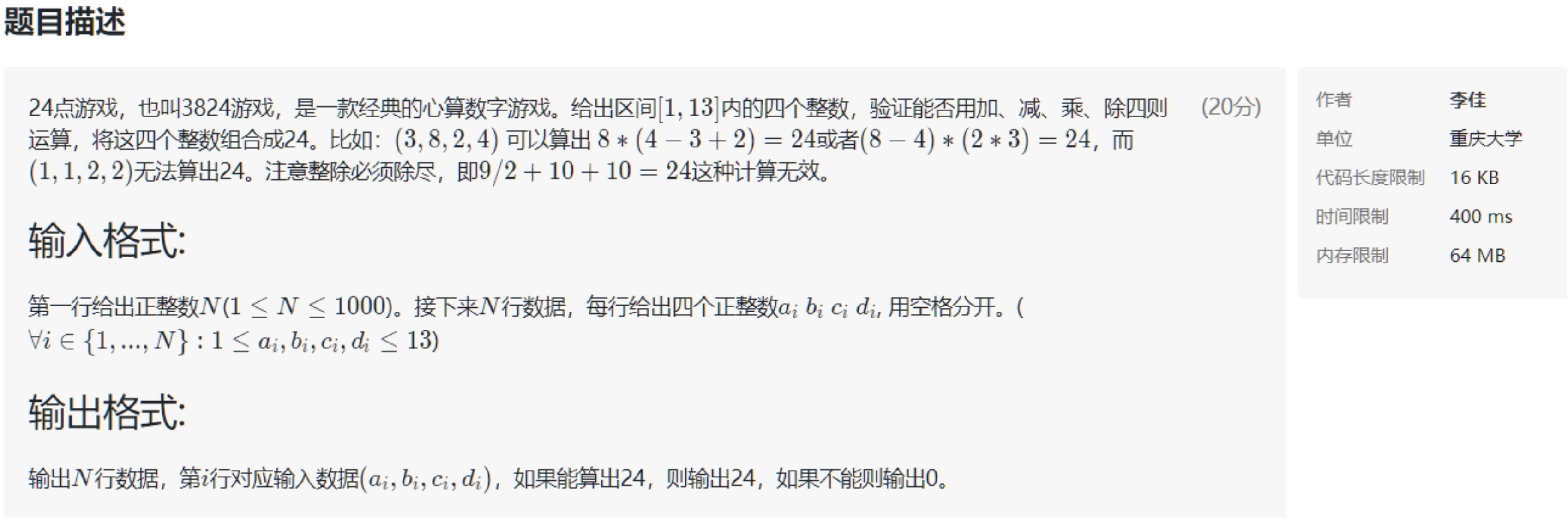
把十位与个位分开，用表格列举所有质数

The Need for Data Structures

- More complex problems demand more computation, making the need for efficient programs even greater.
- A data structure is any data representation and its associated operations.
- Using the proper data structure can make the difference between a program running in a few seconds and one requiring many days.
- A solution is said to be **efficient** if it solves the problem within the required **resource constraints(space and time)**.

The Need for Data Structures

- 复杂问题实例：



题目描述

24点游戏，也叫3824游戏，是一款经典的心算数字游戏。给出区间$[1,13]$内的四个整数，验证能否用加、减、乘、除四则运算，将这四个整数组合成24。比如：$(3,8,2,4)$ 可以算出 $8*(4-3+2)=24$或者$(8-4)*(2*3)=24$，而$(1,1,2,2)$无法算出24。注意整除必须除尽，即$9/2+10+10=24$这种计算无效。 (20分)

作者	李佳
单位	重庆大学
代码长度限制	16 KB
时间限制	400 ms
内存限制	64 MB

输入格式:

第一行给出正整数$N(1 \le N \le 1000)$。接下来N行数据，每行给出四个正整数$a_i\ b_i\ c_i\ d_i$, 用空格分开。($\forall i \in \{1,...,N\}: 1 \le a_i,b_i,c_i,d_i \le 13$)

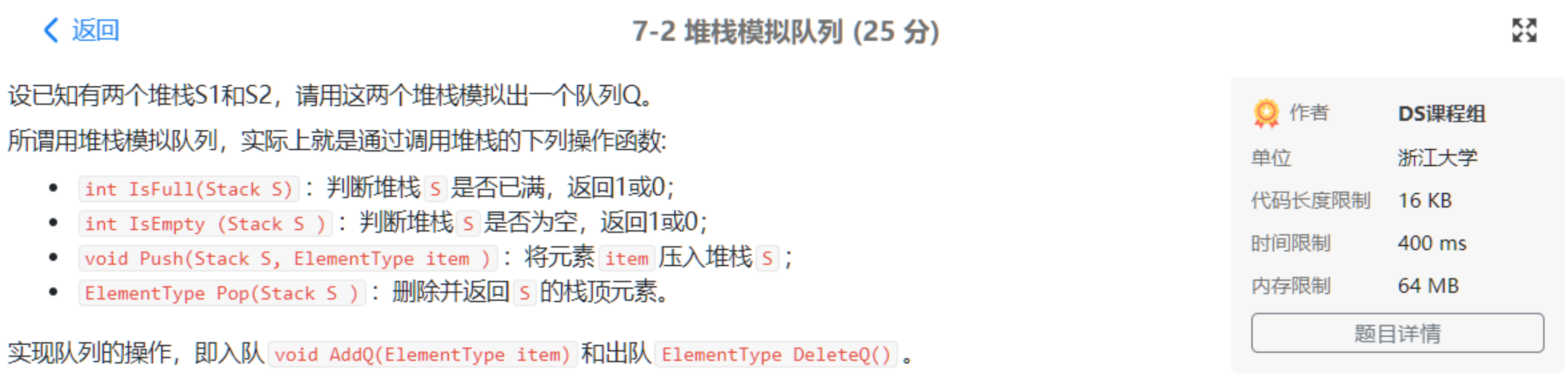
输出格式:

输出N行数据，第i行对应输入数据(a_i,b_i,c_i,d_i)，如果能算出24，则输出24，如果不能则输出0。

需要的数据结构：简单数组
需要的算法：穷举遍历
难度：学完《程序设计基础》课程后能够解决的问题

The Need for Data Structures

- 复杂问题实例：



7-2 堆栈模拟队列 (25 分)

设已知有两个堆栈S1和S2，请用这两个堆栈模拟出一个队列Q。

所谓用堆栈模拟队列，实际上就是通过调用堆栈的下列操作函数:

- `int IsFull(Stack S)`：判断堆栈 `S` 是否已满，返回1或0；
- `int IsEmpty (Stack S )`：判断堆栈 `S` 是否为空，返回1或0；
- `void Push(Stack S, ElementType item )`：将元素 `item` 压入堆栈 `S`；
- `ElementType Pop(Stack S )`：删除并返回 `S` 的栈顶元素。

实现队列的操作，即入队 `void AddQ(ElementType item)` 和出队 `ElementType DeleteQ()`。

作者	DS课程组
单位	浙江大学
代码长度限制	16 KB
时间限制	400 ms
内存限制	64 MB

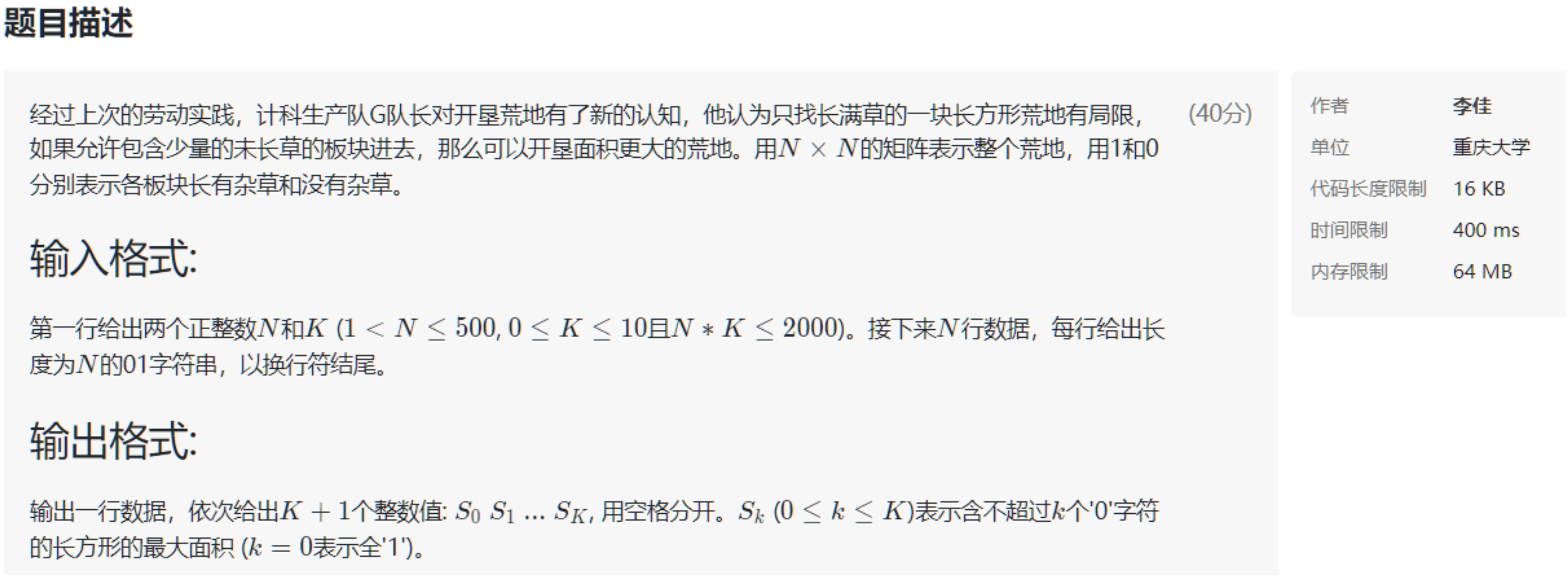
需要的数据结构：堆、栈

需要的算法：栈操作与队列操作的互换

难度：完成《数据结构》课程学习后能够解决的问题

The Need for Data Structures

- 复杂问题实例：



题目描述

经过上次的劳动实践，计科生产队G队长对开垦荒地有了新的认知，他认为只找长满草的一块长方形荒地有局限，如果允许包含少量的未长草的板块进去，那么可以开垦面积更大的荒地。用$N \times N$的矩阵表示整个荒地，用1和0分别表示各板块长有杂草和没有杂草。 (40分)

作者	李佳
单位	重庆大学
代码长度限制	16 KB
时间限制	400 ms
内存限制	64 MB

输入格式:

第一行给出两个正整数N和K $(1 < N \le 500, 0 \le K \le 10$且$N * K \le 2000)$。接下来$N$行数据，每行给出长度为$N$的01字符串，以换行符结尾。

输出格式:

输出一行数据，依次给出$K + 1$个整数值: S_0 S_1 ... S_K, 用空格分开。S_k $(0 \le k \le K)$表示含不超过k个'0'字符的长方形的最大面积 $(k = 0$表示全'1')。

需要的数据结构：行列、单调队列、前缀和数组
需要的算法：动态规划
难度：程序设计竞赛题或知名IT企业面试题

The Need for Data Structures

- 课后练习题（华为招聘面试题）

有二个1到20之间的正整数，甲只知道两个整数之和，乙只知道两个整数的乘积。通过下面的甲和乙的对话，推断这两个数是什么？

甲：我猜不出来。
乙：我也猜不出来！

甲：喔，我猜出来了！
乙：我也知道了。。

问：是什么数？

Picking the best Data Structure

- 1. Analyze your problem to determine the basic operations that must be supported. Examples of basic operations include inserting, deleting and finding.

- 2. Quantify the resource constraints for each operation.

- 3. Select the data structure that best meets these requirements.

Costs and Benefits

- A data structure requires a certain amount of space for each data item it stores, a certain amount of time to perform a single basic operation, and a certain amount of programming effort. Each problem has constraints on available space and time.

- Each solution to a problem makes use of the basic operations in some relative proportion, and the data structure selection process must account for this.

- Only after a careful analysis of your problem’s characteristics can you determine the best data structure for the task.

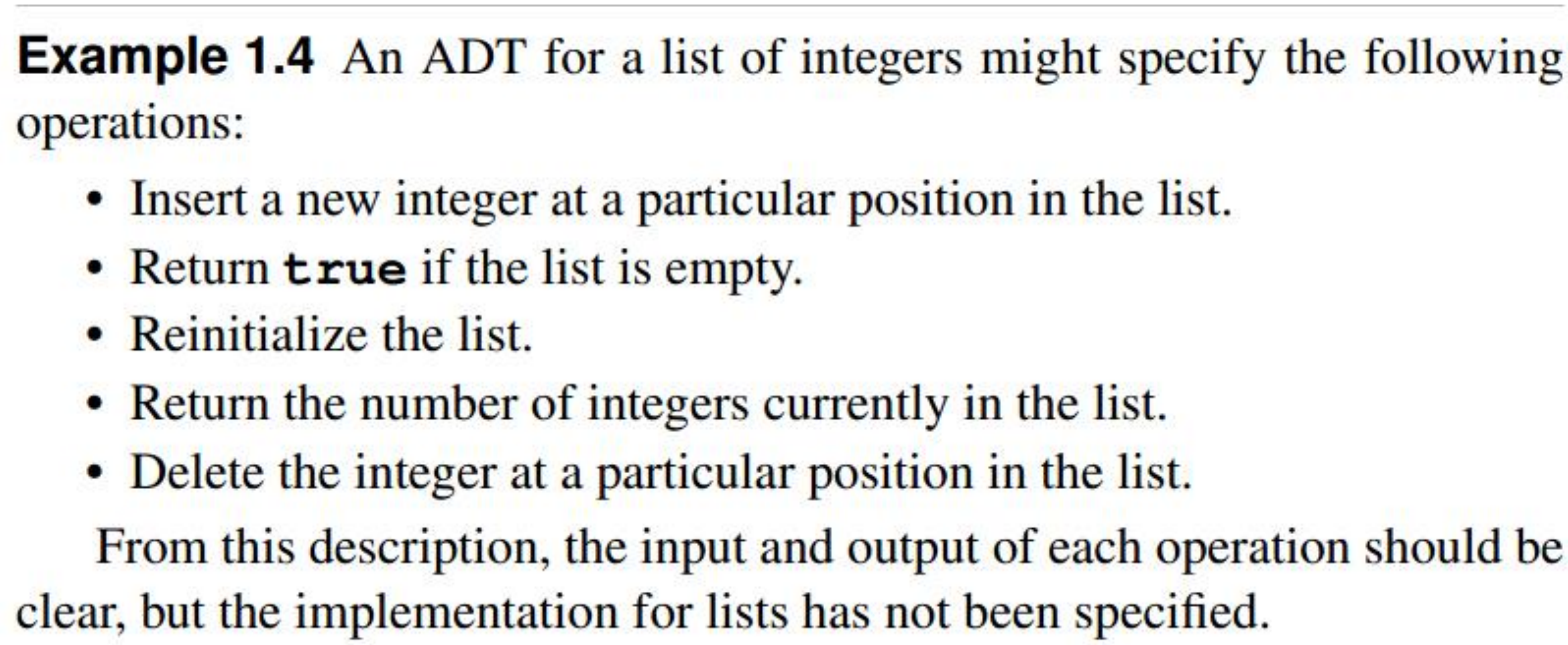
1.2 Terminology

- A **type** is a collection of values .
  - Divided into **Simple Type** and **Aggregate Type** or **Composite Type.**
- A **Data Type** is a type together with a collection of operations to manipulate the type.
- **Data item** is a piece of information or a record whose value is drawn from a type. A data item is said to be a member of a type.
- A distinction should be made between the logical concept of a data type and its **physical implementation** in a computer program.

Terminology

- An **abstract data type(ADT)** is the realization of a data type as a software component. An ADT does not specify how the data type is implemented.
- A **data structure** is the implementation for an **ADT**.
- A given ADT might be supported by more than one implementation.

ADT Example



Example 1.4 An ADT for a list of integers might specify the following operations:

- Insert a new integer at a particular position in the list.
- Return `true` if the list is empty.
- Reinitialize the list.
- Return the number of integers currently in the list.
- Delete the integer at a particular position in the list.

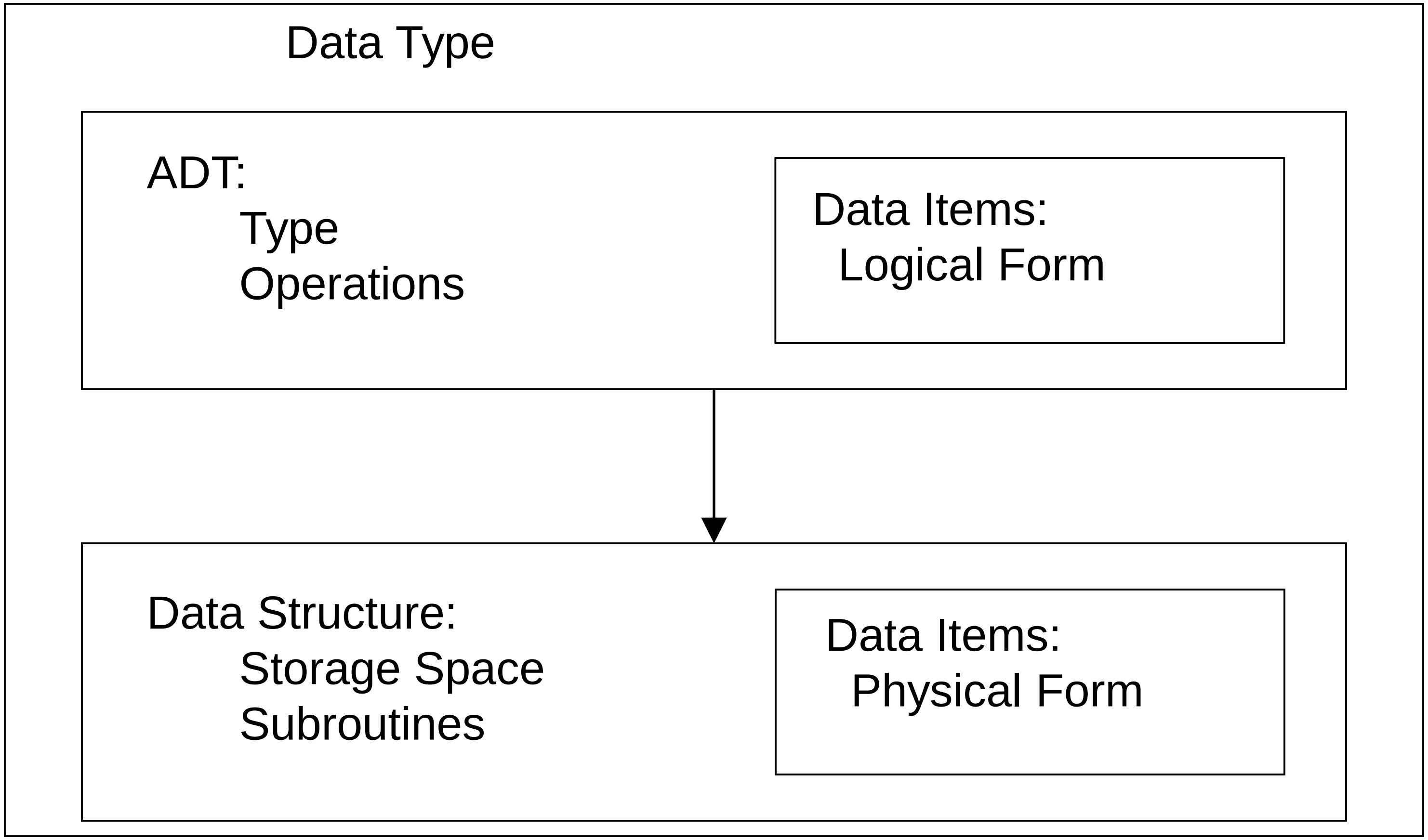
From this description, the input and output of each operation should be clear, but the implementation for lists has not been specified.

Other ADT Examples

- Data structures
  - Lists, Stacks, Queues
  - Heaps
  - Binary Search Trees
  - AVL Trees
  - Hash Tables
  - Graphs
  - Disjoint Sets

Logical vs. Physical Form

- Data types have both a **logical** and a **physical** form.
- The definition of the data type in terms of an ADT is its **logical form**.
- The implementation of the data type as a data structure is its **physical form**.



Data Type

ADT:
Type
Operations

Data Items:
Logical Form

Data Structure:
Storage Space
Subroutines

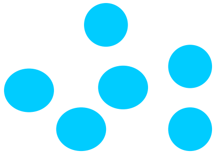
Data Items:
Physical Form

1.3 Topics of Data Structure

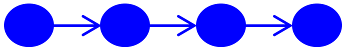
- Logical structure
  - The logical relationship between data.
- Storage structure
  - Storage mapping of logical structure in storage.
- How to implement operations based on above logical and storage structure

Basic Logical Structures

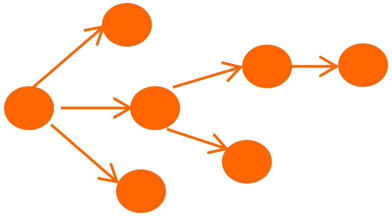
- Set structure



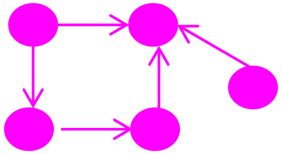
- Linear structure



- Tree structure



- Graph structure



1.4 Problem ,Algorithm and Program

- **Problems**: a problem is a task to be performed.
  - A problem definition should include constraints on the resources that may be consumed by any acceptable solution.

- **Algorithms**: An algorithm is a method or a process followed to solve a problem.
  - If the problem is viewed as a function, then an algorithm is an implementation for the function that transforms an input to the corresponding output.

- **Programs**: A computer program is thought as an instance, or concrete representation, of an algorithm in some programming language.

properties of Algorithm

- 1. It must be **correct**.
- 2. It is composed of a series of **concrete steps**.
- 3. There can be **no ambiguity** as to which step will be performed next.
- 4. It must be composed of **a finite** number of steps.
- 5. It must **terminate**.

Knowledge Points

- Chapter1,pp.3-12,16-18

-End-