市赛前第一次培训讲义

An-Li Alt Ting

September 23, 2014

Contents

1 序

本培训的宗旨是增进受训学生的资讯相关能力，并于参加台北市与国家资讯能力竞赛时，为选手本身与学校争取荣誉。

本培训的目标是让建中校队包办市赛一二等奖。

资讯科学博大精深，资讯科技日新月异，要在如此有限的时间中完整的介绍是很难的。本培训只会以正确为前提，简短介绍（甚至只有提及）与相关竞赛接近的知识。不够详尽的部分就请读者自己查资料了。

本培训研究的知识，主要以讲师的认知为主。然而，为顾及比赛成绩，会特别讲述讲师的认知与环境主流之差异。另外，若有谬误，也请不吝指正。

本次培训主要在电脑教室四举办。在教室内不饮食。不打扰不想被打扰的人学习。

2 时间轴

09-16 Tue 建中校内资讯学科能力竞赛：取前 22 名参加校队选训。

09-22 Mon 市赛前第一次培训：序、时间轴、计算机科学的知识系统、常见题型、常见工具与资源、常见竞赛、计算复杂度理论、计算机的数据表示、计算机的组成。

09-29 Mon 市赛前第二次培训：阵列、堆叠、队列、链接串列。二分搜寻法、排序算法、深度优先搜索、广度优先搜索。C 概论（输入输出（档案）、指标）。

10-06 Mon 市赛前第三次培训：二元搜寻树（线段树）、堆积、并查林。图论常见问题与算法（TS、SP、MST、CC）。C++ 概论（输入输出（档案）、类别）。

10-09 Thu 市赛校队选试：原定两位校队成员，其余二十人选十二人，共十四人代表建中参加市赛。

10-13 Mon 市赛前第四次培训：枚举、递归、贪婪、D&C、DP。网络概论。

10-20 Mon 市赛前第五次培训：集训第一天。数论、排列组合、计算几何、线性代数、抽象代数。操作系统概论。

10-27 Mon 市赛前第六次培训：模拟。竞赛技巧。程设概论。C++ STL 概论。模拟市赛。

10-31 Fri 市赛。

11-03 Mon 国赛前第一次培训：进阶规划、字串算法。

11-10 Mon 国赛前第二次培训：进阶搜寻、进阶图论（2-SAT、Flow、欧拉路径、二分图匹配）。

11-17 Mon 国赛前第三次培训：组合博弈、高级资料结构。

11-24 Mon 国赛前第四次培训：魔鬼训练。

12-01 Mon 国赛前第五次培训：魔鬼训练。

12-08 Mon 国赛前第六次培训：集训第一天。魔鬼训练。

12-15 Mon 国赛前第七次培训：解题通论。

12-22 Mon 国赛（约于此时）。

2015-03-xx TOI 入营考。

2015-04-xx TOI 一阶模考：选十到十二人进到二阶。

2015-05-xx TOI 二阶模考：选四人作为国手。

2015-07-xx IOI 2015

3 计算机科学的知识系统

知识如何分布，事实上取决与人如何理解它。在一个博大精深，关系错综复杂的科学中，分类不再是为了分治，而是构造一个理解它的途径。

为了本次培训，笔者对计算机科学另外作了一次分类，没有提到的部分基本上不讨论。注意到：如果要完整的学习计算机科学，这份分类是行不通的。

以下是本次培训的分类方式：

- 计算理论
  - 算法分析
  - 计算复杂度理论
- 常见设计範式

  - Divide and conquer (D&c) 分治法
    - D&C on Tree 樹分治
  - Dynamic Programming (DP) 动态规划
  - Greedy 贪婪法
  - Back Tracking 试错法
- 常见问题与对应算法
  - Knapsack Problem 背包問題
  - Inversion 逆序數對
  - Longest Common Subsequence (LCS) 最長共同子序列
  - Longest Increasing Subsequence (LIS) 最長遞增子序列
  - Longest Palindromic Substring (LPS) 最長回文子字串
  - Matrix Chain Multiplication 矩陣鏈乘積
  - Multiplication Inverse Element (Modular) 關於模的乘法逆元素
  - Topological Sort (TS) 拓撲排序
  - Single-Source Shortest Path (SSSP) 单源最短路径
    - Dijkstra's Algorithm
    - Bellman-Ford Algorithm
  - All-pairs Shortest Path 全点对最短路径
    - Floyd-Warshall Algorithm
  - Minimum Cost Spanning Tree (MST) 最小成本生成树
    - Prim's Algorithm
    - Kruskal's Algorithm
    - Sollin's Algorithm
  - Connected Component (CC) 连通分量
    - Tarjan's Algorithm
    - Kosaraju's Algorithm
  - Eulerian path 歐拉路徑
    - Hierholzer's Algorithm
    - Fleury's Algorithm
  - Bipartite Matching 二分图匹配
    - Hungarian Algorithm 匈牙利算法
  - Network Flow 网络流
    - Ford-Fulkerson Algorithm
    - Edmonds-Karp Algorithm
  - Lowest Common Ancestor (LCA) 最低共同祖先
    - Tarjan's Algorithm
  - 2-Satisfiability (2-SAT)

  - Minimum Weight Circuit 最小圈
  - Minimum Ratio Cycle 最小比率環
  - Minimum Mean Cycle 最小平均值環
  - Computational Geometry 计算几何
    * Closest Pair 最近點對
    * Farthest Pair 最遠點對

- 常见算法
  - Bubble Sort 气泡排序
  - Quick Sort 快速排序
  - Merge Sort 合并排序
  - Binary Search 二分搜寻
  - Ternary Search 三分搜寻
  - Depth-first Search (DFS) 深度優先搜索
  - Breadth-first Search (BFS) 廣度優先搜索
  - Bidirectional Search
  - A* A 星
  - Iterative Deepening A* (IDA*)
  - 爬山算法
  - 遗传算法
  - Simulated Annealing 模擬退火
  - 禁忌演算法
  - Knuth-Morris-Pratt Algorithm (KMP)
  - Manacher's Algorithm
  - Aho-Corasick Algorithm

- 常见数据结构
  - Array 阵列
  - Stack 堆栈（堆叠）
  - Queue 队列（佇列）
  - Deque 双端队列（双向佇列）
  - Linked List (List) 连结串列（串、链结）
  - Sparse Table 稀疏矩陣
  - Binary Search Tree (BST) 二元搜寻树
    * Segment Tree 线段树
    * Treap 树堆
  - Heap (Priority Queue) 堆积（优先队列）
  - Binary Indexed Tree (BIT) 二分索引樹

  - Union-Find Forest (Disjoint-sets) 併查林（并查集）
  - Trie 前綴樹（字典树）
- 常见编程语言及相关理论
  - 编译器与直译器
  - 编程语言
- 计算机的数据表示
  - 位元
  - 整数的表示方法
  - 字符的表示方法
- 计算机的组成
  - 硬件的设计
  - 运算方法
  - 存储系统
  - 中央处理器
- 网络概论
  - 常见会话与协定
  - 资讯安全问题
  - 云端运算
- 操作系统概论
  - UNIX 与 UNIX-LIKE
  - MS-DOS 与 MS Windows

另外补述，要学好部分的算法，数论、图论、排列组合、线性代数、抽象代数、几何会是重要的数学基础。

4 常见题型

台湾地区常见的题型有笔试题与上机题。

5 常见工具与资源

台湾地区上机测验主要使用 C++ 作为程式语言，编译器以 GCC 为主流。

5.1 编辑器

这里只介绍 Vi 与 Sublime Text 两种编辑器。但是需要注意的一点是：这两种编辑器在比赛环境很有可能是找不到的。

5.1.1 Vi/Vim/gVim

Vi 是一个文字编辑器。它的特色是尽量使用最少的操作达成目标，并且维持双手不离开逐键盘区。它常见于 UNIX 与 UNIX-LIKE 系统中。在很低阶的环境中就可以完整运行。它的学习成本可能是常见编辑器中最高的。Windows 的使用者可以在 Vi 的官方网站找到安装资源。UNIX-LIKE 系统中通常原本就有 Vi，Arch Linux 与 Ubuntu 在 official packages 中就可以找到 Vim 与 gVim。
笔者是使用 Vim 处理几乎所有文书的。

5.1.2 Sublime Text

Sublime Text 是一个文字编辑器。听说有许多高级的功能。
不过，笔者没有实际使用过它。

5.2 IDE

喜欢 IDE 与不喜欢的人都很多，笔者属于后者。这里只介绍 Code::Blocks 与 Dev-C++ 两种 IDE，这两种 IDE 在台湾地区的资讯竞赛是很容易找到的。

5.2.1 Code::Blocks

在 Windows 上有良好支援，在 UNIX-LIKE 上也是。Windows 与 Mac OS X 使用者可以在官方网站上找到安装资源。

5.2.2 Dev-C++

只在 Windows 有支援。原版已经停止开发了，但是很多分支版本。风评基本上都比 Code::Blocks 差。

5.3 书籍与网络资源

5.3.1 Introduction to Algorithms

译有中文版「算法导论」。

5.3.2 培养与锻炼程式设计的逻辑脑 - 秋叶拓哉，岩田阳一，北川宜稔

一本讲竞赛的日本书。错误有些多，但是内容大多容易理解。

5.3.3 名题精选百则（技巧篇）- 冼镜光

讲述了很多演算法的小技巧的书，内容十分有趣，耐人寻味。

5.3.4 算法之道

讲解透彻，内容富有哲理。

5.3.5 C++ Primer

5.3.6 C++ Standards and Drafts

ISO/IEC 14882:1998 First edition

ISO/IEC 14882:2003 Second edition

ISO/IEC 14882:2011 Third edition

N3242: A draft for the 2011 edition.
`http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2011/n3242.pdf`

5.3.7 www.cplusplus.com

有 tutorials 与 references。STL 部分有说明 complexity。
`http://www.cplusplus.com/`

5.4 常见评测系统

5.4.1 TIOJ INFOR Online Judge

网址：`http://tioj.ck.tp.edu.tw/`
建中资讯社的 OJ，由蕭從恩、陳家陞、江懿友与丁安立开发。
继承前 TIOJ 题库。现在建中资讯校队主要使用此 OJ 办比赛。
笔者主观认为题目质量参差不齐。

5.4.2 Codeforces

网址：`http://codeforces.com/`
经常有线上赛。可以看到别人的 source code。

5.4.3 HSNU Online Judge (HOJ)

网址：`http://hoj.twbbs.org/judge/`
师大附中的 OJ，有许多 OI 题。

5.4.4 PKU Online Judge (POJ)

网址：http://poj.org/
北京大学的 OJ，题目量多，网络上有流传良好题单。

5.4.5 Saratov State University :: Online Contester

网址：http://acm.sgu.ru/
笔者主观认为题目质量高。

5.4.6 STEP5 Online Judge

网址：http://web2.ck.tp.edu.tw/~step5/
2012-2014 年间，建中校队使用的 OJ。

5.4.7 UVa Online Judge

网址：http://uva.onlinejudge.org/
老牌的 OJ。有良好的题目管理系统。

5.4.8 Zero Judge

网址：http://zerojudge.tw/
收录了许多台湾地区的竞赛题目。

6 常见竞赛

6.1 台北市高中资讯学科能力竞赛

本次培训的目标竞赛。一般在师范大学举办。笔试成绩占 30%，上机成绩占 70%。
题目容易出问题。

6.2 全国高中资讯学科能力竞赛

一般由清华大学或交通大学举办，比两天。题目容易出问题。

6.3 National Problem Solving Contest on Internet (NPSC)

教育部委托台大办的比赛。今年无需报名费。国科会举办时有丰厚奖品。
2014 年官方网站：http://contest.cc.ntu.edu.tw/npsc2014/

6.4 International Olympiad in Informatics (IOI)

资讯奥林匹亚竞赛，高中生算法竞赛的最高荣誉。

7 算法分析

算法分析（analysis of algorithms）。

7.1 成长量级

成长量级（order of growth）。

7.1.1 渐近符号

对于实数函数 $f(x)$ 来说，$O(f(x))$ 是一个函数的集合，包含所有当 x 趋近于无限大时，成长行为至多与 $f(x)$ 一样快的函数。这里说的快是一种级别，一般认为 $f(x)$ 的线性函数的成长行为与 $f(x)$ 都是一样的。

形式化定义：$f(x) \in O(g(x)) \equiv \exists\ x_0, c > 0 : |f(x)| \leq c|g(x)|\ \forall\ x > x_0$

$O(\ldots)$ 这种表示方法称为大 O 符号（Big O notation）。相似的 Ω、Θ、o、ω 都是渐近符号。其定义为：

$f(x) \in \Omega(g(x)) \equiv g(x) \in O(f(x))$
$f(x) \in \Theta(g(x)) \equiv f(x) \in O(g(x)) \wedge g(x) \in O(f(x))$
$f(x) \in o(g(x)) \equiv \lim \frac{f(x)}{g(x)} = 0$
$f(x) \in \omega(g(x)) \equiv g(x) \in o(f(x))$

我们经常用 $\Theta(f(x))$ 或 $O(f(x))$ 代表一个函数阶，而且 $f(x)$ 通常是仅有领导项而领导系数为 1 的函数。下表列出一些常见的函数阶：

符号	名称	意义及其代表问题或算法
$\Theta(1)$	常数时间	二进位整数判断奇偶性。
$\Theta(\log n)$	对数时间	二分搜寻法能有效解决的问题。
$\Theta(n)$	线性时间	两整数相加。
$\Theta(n \log n)$	线性对数时间	比较排序。
$\Theta(2^n)$	指数时间	3-SAT Problem。
$\Theta(n!)$	阶乘时间	销售员旅行问题。

值得注意的是 $\Theta(0)$ 是存在的。

其中，所有属于 $O(n^c)$（$c \in \mathbb{N}$）的时间函数，又称多项式时间。

7.2 摊销分析

8 计算复杂度理论

计算复杂度理论（computational complexity theory），是一个主要研究解决一个可计算问题时所需要的资源的理论，是计算理论与数学的分支。

讨论复杂度时，我们会需要定义一个计算模型，一个计算机的抽象形式，以及与其相关的计算资源。计算模型通常是图灵机或电路；计算资源通常至少有时间与空间，每执行一个步诹都花费其对应的时间并有可能存取部分的空间，单位时间与单位时间下的单位空间都不能被重复利用。值得注意的是，在台湾高中生的程式竞赛中，更容易以 x86 处理器上的 C 讨论。

8.1 图灵机

图灵机（Turing Machine），或称为确定型图灵机（Deterministic Turing Machine），是一个计算模型。是由英国数学家图灵于 1936 年所提出的抽象计算模型。它等价于任何在确定且有限的步诹内完成目标计算的机器。

通用图灵机（Universal Turing Machine）是指一个有能力藉由输入任何一个图灵机的代码，完全模仿该图灵机的图灵机。当今普遍作为 PC 处理器的 x86 就是通用图灵机的一个实例。

作为一个图灵机的变体，非确定型图灵机（Non-deterministic Turing Machine）是一种可以任意转移到其他状态的机器。非确定型图灵机是否存在，至今还是哲学与科学里争论不休的问题，不过就物理学观点，确定型图灵机与非确定型图灵机是很难同时存在的。

8.2 竞赛里常见的复杂度类

P、NP、NP-hard、NP-complete（NPC）都是竞赛里常见的复杂度类。

一个问题 L 属于 P（polynomial time、PTIME、DTIME($n^{O(1)}$)）若且唯若 L 可以用确定型图灵机在多项式时间内解决。

一个问题 L 属于 NP（non-deterministic polynomial time）若且唯若 L 是一个判定性问题，且可以在多项式时间内被验证。

一个问题 L 属于 NP-hard 若且唯若任何属于 NP 的问题 H，都可以在多项式时间内转化为 L。

NP-compelete 是 NP 与 NP-hard 的交集。

其中 P⊂NP，NPC⊂NP，NPC⊂NP-hard。

另外，我们已经知道，但是我们还不知道是否 P=NP。

9 计算机的数据表示

9.1 位元

位元（bit）是主流电子计算机唯一的最小资料单位。

在主流电子计算机中总是使用 bit 作为唯一的资料单位的原应有二：

一、在物理上容易实现：要找到有两种稳定态的物理器件，相较于找到有三种以上的稳定态的物理器件来说，是相对容易很多的。

二、容易建构逻辑电路，并以逻辑电路构造基本代数运算：两种状态可以分别对应假/眞，这样方便建构逻辑电路；而这两种状态也可以分别对应二进制数的两种数字 0/1，二进位的加减乘除也是相对容易实现的。

一组 bits 被称为位元组（byte、字节）。byte 的实际长度没有统一的定义，但通常是 byte=8bits。

若干个 bytes 成字（word），而字的长度取决于计算机的字长。例如 64 位元的机器，word=64bits。

IEEE 1541-2002 建议可将 bit 简写为小写字母 b，而 byte 则简写为大写字母 B。

位元或位元组以十为底的次方倍数单位也可以前缀国际单位制词头简写，而 IEEE 1541-2002 也有定义以二位底的次方倍数词头：

值	名称	符号	值	名称	符号
1000	k	kilo	1024	Ki	kili
1000^2	M	mega	1024^2	Mi	mebi
1000^3	G	giga	1024^3	Gi	gibi
1000^4	T	tera	1024^4	Ti	tebi
1000^5	P	peta	1024^5	Pi	pebi
1000^6	E	exa	1024^6	Ei	exbi
1000^7	Z	zetta	1024^7	Zi	zebi
1000^8	Y	yotta	1024^8	Yi	yobi

例如 1000bits=kb，1000bytes=1MB，1024bytes=1MiB，以此类推。

9.2 整数的表示方法

计算机中几乎总是以二进位表示数值。

在计算机中表示整数时，通常会先指定该整数的位元长（bit-length)，表示在记忆体中使用多少的位元来记录这的整数。

由组合数学可知，位元长为 n 的整数，可以表达 2^n 种不同的值。

9.2.1 无号整数

无号整数是指没有正负号的整数，也就是非负整数。它表示方法与数学上的表示方法，在精神上基本相同。但是因为有指定位元长，所以位元长为 8 的十进位整数 13，会被表示为：00001101。

9.2.2 有号整数

有号整数就是一个正号或负号，接上一个无号整数。一般来说，表示有号整数时，我们会舍弃最高位的权值，以最高位的位元来表示正负号。通常，又以 0 表示正号，以 1 表示负号。

以令开头的数 0，所表达的数，基本上与无号整数相同。而关于以 1 开头的数，则有三中常见的解释方法，它们分别是原码、反码、补码。

一个位元长为 n 的有号整数 m，在记忆体中被表示为字符串 s，有以下三种解释方式：

一、原码：s 最低的 $n-1$ 位表示 m 的绝对值。例如 s=10000111 表示 $m=-7$；s=00000000 表示 $m=+0$；s=10000000 表示 $m=-0$。

二、反码：m 与 $-m$ 的相反数的每一位都是相反的。例如 s=00000101 表示 $m=+5$；s=11111010 表式 $m=-5$；s=00000000 表示 $m=+0$；s=11111111 表示 $m=-0$。

三、补码：m 在 $\bmod 2^{n-1}$ 下与 s 作为无号整数时所表示的整数同余。例如 s=00000110 表示 $m=+6$；s=111111010 表式 $m=-6$；s=00000000 表示 $m=+0$；没有表示 $m=-0$ 的 s。

9.2.3 常用进位制

二进制、八进制、十六进制都是计算机里常用的进位制。

二进制通常用于记录资料。八进制有时取代十六进制。十六进制经常用于表示记忆体位址或色码。

二进制、八进制、十六进制之间的转换是不需要除法的，除了表示起来符合空间上的直觉外，也加快了转换的效率。

在 C 的整数值符（integer literal）中，前缀 0x 的表示十六进位数；否则，前缀 0 的整数值符表示八进位数；否则表示十进位数。

9.2.4 相关数学

欲求一数 n 的 k 进位表示：

$n = q * k^m + r$，其中 $k^m \leq n \wedge m \in \mathbb{N}$，$q$ 的 k 进位表示接上 r 的 k 进位表示，就是 n 的 k 进位表示。

短除法与权值法都依赖这个原理，正着用就是短除法，反着用就是权值法。

9.3 字符的表示方法

9.3.1 ASCII

American Standard Code for Information Interchange (ASCII、美国信息交换标准代码)。定义了 7-bit integer 所对应的 128 个字元，包含控制字元、阿拉伯数字、大小写英文字母以及标点符号。

9.3.2 Unicode

Unicode（萬國碼、國際碼、統一碼、單一碼）。是一个整理了世界上大部分的文字系统的编码系统。

9.3.3 BIG5 与 GBK

BIG5 又称大五码。1984 年，由台湾五家软件公司创立，因而得名。创立动机是统一当时不同厂商推出的编码，以及解决 GB2312 没有繁体字的问题。

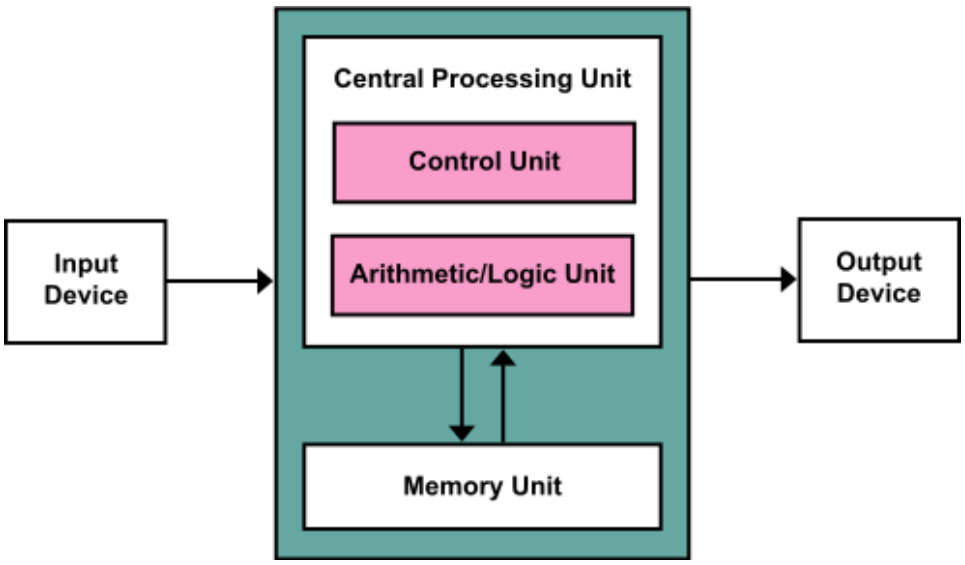
GBK 是中共在 1995 发布的中文编码扩展国家标准。提供了 1894 个造字位码。兼容了繁体与简体。

10 计算机的组成

现代一台完整实用的通用计算机，由硬体与软体两部分组成。以纸本文书来比喻，硬体是纸张与油墨，软体就是油墨在纸张上面的分布，所体现的文字模式。

10.1 硬件的设计

现代的计算机多依据冯·诺伊曼体系结构（von Neumann architecture）设计而成。



10.2 运算方法

基础的运算主要分为两类：逻辑运算与代数运算。
逻辑运算包括：非、或、与、异或。
代数运算包括：加、减、乘、除。

在加、减、乘时，可能会有溢位的状况发生。

一般来说，整数的除法被分为：求商与模除。

在笔试题目中，逻辑运算经常以布尔代数、布尔环、C 的位元算子、逻辑算子的形式出像。

10.3 存储系统

现代计算机的资料主要存在四种装置上：CPU 的 register（暂存器）、CPU 的 cache（高速缓存）、RAM（Random-access memory 随机存取存储器）、外部存储器（硬盘、光盘、磁片）。其中校能由先到后递减，容量由先到后递增。

Register、cache 与 RAM 只有在通电时才能保留资料。

其中 RAM 又分为 SRAM (Static Random-Access Memory) 与 DRAM (Dynamic Random Access Memory)。

10.4 中央处理器

中央处理器（Central Processing Unit，CPU）。被做为单片集成电路的 CPU 又被称为微处理器（Microprocessor）。它的功能包括程序控制、操作控制、时间控制、数据加工。最基本的 CPU 由控制器与运算器组成。

CPU 中至少要有六类的寄存器：数据寄存器（Data Register，DR）、指令寄存器（Instruction Register，IR）、程序计数器（Program Counter，PC）、地址寄存器（Address Register，AR）、累加寄存器（Accumulator，AC）、程序状态字（Program Status Word，PSW）。

11 References

Wikipedia
http://www.wikipedia.org/

Analysis of algorithms - Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Analysis_of_algorithms

Computational complexity theory - Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Computational_complexity_theory

Von Neumann architecture - Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Von_Neumann_architecture

计算机组成与系统结构 - 陈泽宇主编
http://share.onlinesjtu.com/pluginfile.php/1605/mod_resource/content/1/%E8%AE%A1%E7%AE%97%E6%9C%BA%E7%BB%84%E6%88%90%E4%B8%8E%E7%B3%BB%E7%BB%9F%E7%BB%93%E6%9E%84.pdf