

# 大湾区青少年信息学编程认证大纲

(Big Bay Computer Certification Outline)

(2022 版)

深圳市计算机学会

2022 年 2 月

# 前言

“计算机普及要从娃娃抓起”——邓小平。编程是信息技术的重要组成部分，但学习编程的核心，不在于掌握哪一种计算机语言——编程语言在不断革新，在今天几行看似炫酷的代码，在不久的将来一定会成为老掉牙的古董。学习编程的本质，实际在于思维方式的养成，这种计算机性的思维方式，即 Computational Thinking，被称作计算思维。

教育部部长怀进鹏：以教育信息化推动教育高质量发展，以教育信息化引领教育现代化。2017年，在《浙江省深化高校考试招生制度综合改革试点方案》中，信息技术被正式列入高考选考科目，信息技术第一次登上高考舞台。同年，国务院出台《新一代人工智能发展规划》，从国家高度，明确指出信息技术教育的重要性，号召“在中小学阶段设置人工智能相关课程，逐步推广编程教育”。在可预见的未来，信息技术教育将成为基础教育的重要部分。

美国卡内基·梅隆大学计算机科学系主任周以真(*Jeannette M. Wing*)教授在美国计算机权威期刊《Communications of the ACM》杂志上，将计算思维定义为：运用计算机科学的基础概念进行问题求解、系统设计、以及人类行为理解等涵盖计算机科学之广度的一系列思维活动。掌握计算思维，也就掌握了解决问题的能力。

深圳市计算机学会联合澳门电脑学会、香港电脑教育学会，为培养青少年计算思维，提高青少年认识问题、处理问题的能力，共同举办大湾区青少年信息学编程认证（简称 BBCC）。BBCC 认证拥有四大创新“理念创新、赛制创新、体制创新、机制创新”。跨地域、跨学科、在线认证，中英双语，立体多维赛制，层级式迭代比赛体系，未来将面向全球 5 大湾区（粤港澳、硅谷、纽约、东京、伦敦）走向国际赛事。2020 年，第一届 BBCC 赛事成功举办，吸引来自全国四千五百余名青少年报名。2021 年，第二届 BBCC 在认证模式和活动形式上采取众多创新举措，欢迎您查阅大湾区青少年信息学编程认证手册，了解更多内容。

2022 年 2 月

本大纲版权归 BBCC 组委会所有，如有抄袭串改等行为，将依法追究法律责任

# 目录

一、 简介.....	4
(一) 大湾区青少年信息学编程认证手册.....	4
(二) 大湾区青少年信息学编程认证.....	4
(三) 粤港澳互联网联盟.....	5
(四) 深圳市计算机学会.....	6
(五) 澳门电脑学会.....	7
(六) 香港电脑教育学会.....	7
二、 大纲.....	8
BBCC 代码编程知识点.....	8
1. BBCC CP U12 组认证要点.....	8
1.1 能力目标.....	8
1.2 BBCC CP I (U12 初级) 知识点.....	8
1.3 BBCC CP II (U12 高级) 知识点.....	12
2. BBCC CP U15 组认证要点.....	14
2.1 能力目标.....	14
2.2 BBCC CP III (U15 初级) 知识点.....	14
2.3 BBCC CP IV (U15 高级) 知识点.....	15
3. BBCC CP U18 组认证要点.....	16
3.1 能力目标.....	16
3.2 BBCC CP V (U18 初级) 知识点.....	16
3.3 BBCC CP VI (U18 高级) 知识点.....	18
附表 1-1 BBCC U12 组(BBCC CP I/II)代码编程学习指南.....	20
附表 1-2 BBCC U15 组(BBCC CP III/ IV)代码编程学习指南.....	22
附表 1-3 BBCC U18 组(BBCC CP V/VI)学习指南.....	23
附表 2 BBCC 等级对照表(代码编程部分).....	25
附录 1 代码编程知识点索引.....	26
联系我们.....	27

## 一、简介

### (一) 大湾区青少年信息学编程认证手册

- 1、手册旨在为大湾区青少年信息学编程认证(BBCC)的题目命制提供依据。
- 2、手册向参加 BBCC 的学生和信息学爱好者提供学习参考范围。
- 3、大纲部分依照 BBCC I-VI 级，划分六级知识点。代码编程知识点涵盖 C++、Python 两种语言。
- 4、高等级认证将自动包含低等级认证的所有知识点。

### (二) 大湾区青少年信息学编程认证

大湾区青少年信息学编程认证(Big Bay Computer Certification, 简称 BBCC)是由粤港澳大湾区互联网联盟指导, 深圳市计算机学会(SZCCF)、香港电脑教育学会、澳门电脑学会共同主办, ACM SIGCSE China、国际大学生程序设计竞赛(ICPC)组织等十四家国际及地方性学术团体协办的培养 9 岁到 19 岁青少年算法和编程能力的活动。

BBCC 将严格按照国家政府相关政策制定相关的组织、筹备、协调工作, 坚持学术支撑、科学指导; 坚持认证公平、公正、公开的原则; 坚持不以营利为目的, 坚持不与升学挂钩, 不与学科教育成绩挂钩; 遵循自愿参加的原则。

2022 年 BBCC 以国际化标准, 对标 NOIP、USACO、IOI 等信息学权威竞赛, 精准鉴定青少年现阶段的编程能力。选手可选择代码编程(C++、Python)参与 BBCC, 并且一年中可以多次参加, 逐步提升编程级别, 稳步提高编程能力。

### （三）粤港澳互联网联盟

2020年12月8日，粤港澳大湾区互联网联盟成立大会顺利举行，大会推选深圳市计算机学会连同广州、香港、澳门三地代表担任联席理事长单位。

粤港澳大湾区互联网联盟（以下简称“互联网联盟”），在粤港澳大湾区科技协同创新联盟的指导下，为贯彻落实《粤港澳大湾区发展规划纲要》，团结和引领大湾区互联网相关产业工作者投身湾区建设，由粤港澳大湾区互联网等新一代信息技术产业相关社会团体共同发起成立。



互联网联盟将促进粤港澳大湾区优质资源共建共享，搭建多元化沟通交流学习平台，探索人才培引和科技成果转化，推动前沿技术与实体经济有效对接，着力提升粤港澳大湾区互联网等新一代信息技术产业的竞争力。

粤港澳大湾区科技协同创新联盟是中国科协的指导下，由广东省科协、广州市科协、深圳市科协携手香港京港学术交流中心、澳门科学技术协进会共同发起，联合粤港澳大湾区9+2城市致力于推动大湾区科技协同创新的科技团体、高等院校、科研院所、企业以及有关人士自愿组成的公益性、非法人组织。

#### （四）深圳市计算机学会

深圳市计算机学会(SZCCF)是由从事计算机行业的科技工作者，与计算机相关的科研、教育、管理、产业等领域的个人及单位自愿组成的学术性、联合性、专业性的地方性非营利性社会组织。学会业务主管单位为深圳市科学技术协会，受深圳市民政局的业务指导及监督管理。学会已进入深圳市承接政府职能转移和购买服务社会组织推荐目录，当选粤港澳大湾区互联网联盟联席理事长单位。



学会致力于打造具有高端学术影响力的社会组织，坚持搭建高端平台，推动信息科技产学研资源交流，整合多方资源，强化高端人才供应链服务，建立学会自有专家智库，布局大数据、未来网络、人工智能、区块链、量子科技、数字经济、信息安全等计算机领域；依托学会“专家智库”配合政府职能，提供行业资讯服务，为政府决议建言献策，受到广泛好评。

## (五) 澳门电脑学会



澳门计算机学会于1983年成立，初期由一群爱好计算机科技的人士创立，后来经过30年的发展，成为一个主要由从事信息科技工作的专业人士组成的非牟利团体，会员来自公共及私人领域的专业人员及教育工作者。

## (六) 香港电脑教育学会



香港计算机教育学会始创于1981年底，当时香港教育署正计划在中学进行计算机科试验计划，一群热心的老师于是组织起来，创立香港计算机教育学会，以便互相支持，共同推动香港的计算机教育。

## 二、大纲

### BBCC 代码编程知识点

#### 1. BBCC CP U12 组认证要点

##### 1.1 能力目标

	BBCC CP I	BBCC CP II
语言能力	了解程序的语法元素、基本符号	掌握算术运算、关系运算、逻辑运算等运算的高级语言表达
编程能力	能够编写简单的循环、分支结构语句	能够实现分支语句的嵌套、多层循环的构造等
工具能力	能够使用集成开发环境的基本功能	熟悉集成开发环境的常用功能
程序设计能力	能够使用变量、数组等设计程序，能够编写、使用简单函数	能够使用递归函数等解决问题，深入了解函数的使用方法
数学能力	能够编程实现基本算术运算，了解数制及不同数制的相互转换	掌握基础数论的内容

##### 1.2 BBCC CP I (U12 初级) 知识点

###### 1.2.1 计算机基础及原理

###### 1. 计算机历史

- (1) 重要人物及历史事件以及对计算机发展的贡献
- (2) 计算机的发展历史（四代）与计算机的分类

###### 2. 计算机基础与原理

- (1) ASCII 编码的应用与转换

(2) 汉字编码的实现与分类

(3) 计算机内部原码、反码和补码的原理和计算方式

(4) 进制转换与逻辑运算

(5) 计算机中图像的显示（三基色与三原色）、存放形式与存储大小的计算

### 3. 计算机硬件

(1) 冯·诺依曼理论框架下计算机硬件的组成，以及每个部分的功能与作用

(2) 能够区分各种内存和外存，以及其作用和读写速度的快慢

(3) 能够正确区分和认知每个组成部分

### 4. 计算机软件

(1) 能够区分系统软件和应用软件

(2) 理解从用户到硬件之间各种软件的层级关系

(3) 目前主流的操作系统(OS):Windows/Unix/Linux/macOS/iOS

### 5. 计算机语言

(1) 了解程序设计语言的三大类：机器语言、汇编语言和高级语言，并且对其优势和作用有非常清楚的认知

(2) 能够理解和区分解释性语言和编译性语言

### 6. 计算机网络

(1) 网络的发展阶段

(2) 网络的基本概念与分类（作用范围与拓扑结构类型）

(3) OSI 模型与 TCP/IP 模型的内容以及相互对应

(4) 理解 IP 地址、网关、类别以及其计算方式

## 1.2.2 语言概述

1. 编程语言的基本符号
2. 编程语言的语法元素（缩进、注释、关键字、标识符、常量、运算符、标点符号、赋值语句、引用等）
3. 程序的基本框架
4. (Python) 编程语言的基本特性（解释型，交互性，面向对象等）
5. (Python) IDLE 或 Visual Studio Code 集成开发环境的配置、编辑、运行与调试等功能
6. (C++) 掌握 Dev C++ 5.9.2 或 Visual C++ 2010 版集成开发环境的编辑、编译、运行
7. (Python) 能够阅读给出的 Python 代码，并且模拟 Python 程序的运行过程

## 1.2.3 数据类型、表达式和基本运算

1. 基本数据类型（整型、实型、字符型、布尔型）的类型描述、数据范围等
2. 常量定义及引用
3. 变量的定义与使用方法（定义及初始化，全局和局部变量、变量的自增和自减）
4. 运算符的种类、运算优先级和结合性
5. (Python) 基本数据类型（整数、浮点数、字符串）的使用及不同数据类型的判断和转换
6. (Python) 数值运算操作符，数值运算函数，字符串的操作符、索引、切片及基本的 `format()` 格式化方法
7. (Python) 表达式类型及求值规则（赋值运算，算术运算符和算术表达式）

## 1.2.4 程序设计基本语句

1. 赋值语句、表达式语句、复合语句、输入、输出语句和空语句等

2. if、switch 语句实现分支结构
3. for、while、dowhile 语句实现循环结构
4. continue、break 和 return 语句的运用
5. (Python)程序的异常处理的基本概念

### 1.2.5 数组

1. 一维数组的定义、初始化和访问
2. 多维数组的定义、初始化和访问
3. 字符串与字符数组的定义和使用

### 1.2.6 (Python)列表和元组

1. 列表和元组的概念与区别
2. 列表和元组的定义、初始化和访问
3. 列表和元组的索引，截取和基本内置函数

### 1.2.7 (Python)字典

1. 字典的定义，初始化
2. 字典中键/值的访问，修改和删除
3. 字典的基本内置函数和方法

### 1.2.8 函数的有关使用

1. 函数的定义方法和调用方法
2. 函数的类型和返回值
3. 形式参数与实际参数，参数值的传递

### 1.2.9 基础数学

1. 算术运算（加、减、乘、除、整除、取余）
2. 数制（进制概念、十进制、二进制、八进制、十六进制等概念及其相互转换）

## 1.3 BBCC CP II (U12 高级) 知识点

### 1.3.1 语言概述

1. Dev C++ 5.9.2 或 Visual C++ 2010 版集成开发环境的编辑、编译、运行与程序调试等功能

### 1.3.2 数据类型、表达式和基本运算

1. 赋值运算

2. 算术运算符和算术表达式

3. 关系运算符和关系表达式

4. 逻辑运算符和逻辑表达式

5. (C++) 条件运算，逗号表达式

6. (Python) 成员运算符，身份运算符，运算符优先级

### 1.3.3 程序设计基本语句

1. 分支语句的嵌套构造及其运用

2. 二层循环的构造及其运用

3. 多层循环的构造及其运用

### 1.3.4 函数的有关使用

1. 变量的作用域和生存周期

2. 递归函数

3. 常用字符串函数（长度、查找、比较、拷贝、连接、插入、截取、替换等）的使用

### 1.3.5 (Python) 面向对象

1. 基本的面向对象的概念和知识

2. Python 中类的定义和使用方法

3. Python 中类的属性与方法及其应用

#### 4. 理解 Python 中类的继承、多态与方法重载

##### 1.3.6 基础算法

1. 时空复杂度分析（复杂度的概念、时间复杂度估算、空间复杂度估算）
2. 基础算法（模拟、枚举、贪心、递归、前缀和、分治与倍增、二分）
3. 排序（排序概念、冒泡排序、简单插入排序、简单选择排序、sort 函数）

##### 1.3.7 简单数据结构

1. 顺序表的概念及使用
2. 栈的概念与使用
3. 队列的概念与使用

##### 1.3.8 (Python) 计算生态

1. 标准库与基本的 Python 内置函数
2. 第三方库的获取与安装
3. 更加广泛的 Python 的生态，只要求有基本的了解，不要求完全掌握，包括但不局限于 Python 在网络爬虫、数据分析、Web 开发、游戏开发、机器学习、数据可视化等方面的应用

##### 1.3.9 基础数学

1. 基础数论（质数、合数、因数、倍数概念，质数的判断、辗转相除法算法及其应用）

## 2. BBCC CP U15 组认证要点

### 2.1 能力目标

	BBCC CP III	BBCC CP IV
编程能力	能够使用结构体、类，能够应用链表、树、图等数据结构	能使用简单模板库简化编程，能够使用较复杂数据结构
算法能力	能够编写高精度数的运算算法，实现归并排序、快速排序等较复杂算法，了解动态规划的基本原理	能够编程实现分治、倍增等算法，解决集合、划分类型的动态规划等，掌握简单图论算法
数学能力	能够掌握初中代数运算	掌握基础数论，如素数筛、排列组合等

### 2.2 BBCC CP III (U15 初级) 知识点

#### 2.2.1 数据类型和模板

1. (C++) 结构体的声明与使用，运算符重载，结构体数组
2. (C++) 指针运算符 (& 和 \*) 的含义，指针的定义和使用指针与数组，指针与字符数组，结构体指针
3. (Python) 类的声明与使用，类的继承，多态与方法重载
4. (Python) 引用的概念，与 C 语言中指针的区别
5. 文本文件的读写，文件的重定向操作

#### 2.2.2 基础算法

1. 高精度的表示方法与基本运算（加、减、乘和高精度与单精度的取余除法）
2. 基础算法（差分约束系统、尺取法与双指针、离散化思想）
3. 排序算法（基数排序、归并排序、快速排序）

### 2.2.3 动态规划

1. 动态规划的基本思想和基本原理
2. 常见线型动态规划问题，如求最长上升子序列、最长公共子序列等

### 2.2.4 图论算法

1. 简单图上的搜索 (DFS 与 BFS)

### 2.2.5 数据结构

1. 链表：单向链表，双向链表，循环链表的构造与使用
2. 树的概念与表示方式（父亲表示法和孩子表示法）
3. 图的基本概念与存储方式（邻接表与邻接矩阵）

### 2.2.6 基础数学

1. 初中代数运算

## 2.3 BBCC CP IV (U15 高级) 知识点

### 2.3.1 数据类型和模板

1. (C++&JAVA) STL 库：vector、stack、queue、list 等模板的应用
2. (Python) 熟练使用 Python 中的列表，元组与字典，及其常用的内置函数和方法

### 2.3.2 基础算法

1. 基础算法（简单分治、简单倍增、快速幂）

### 2.3.3 动态规划

1. 常见集合类型动态规划问题，如背包问题等
2. 常见划分类型规划问题，如石子合并等

### 2.3.4 图论算法

1. 两种最短路算法——Floyd 算法与 Dijkstra 算法
2. 有向无环图的拓扑排序

### 2.3.5 数据结构

1. 二叉树的概念与存储方式
2. 二叉树的遍历（先序、中序、后序）
3. 哈夫曼树的概念与应用

### 2.3.6 基础数学

1. 基础数论（素数筛、排列组合）

## 3. BBCC CP U18 组认证要点

### 3.1 能力目标

	BBCC CP V	BBCC CP VI
编程能力	能够在编程中应用特殊数据结构（特殊队列、并查集等）	能够在编程中应用特殊数据结构（特殊树、特殊图等）
算法能力	能够实现较简单搜索算法、图论算法，解决状态压缩和树上动态规划问题	能够实现较复杂搜索算法、图论算法、字符串算法，实现动态规划的一系列常用优化
数学能力	能够掌握高中代数运算，解析几何、立体几何、线性代数的基本运算、初等数论	能够使用组合数学定理和一些重要数学定理解决问题

### 3.2 BBCC CP V (U18 初级) 知识点

#### 3.2.1 语言特性

1. (C++&Python) 类(class) 的声明与使用
2. (Python) 常用库(time、random、math、re) 的原理和使用

#### 3.2.2 基础算法

1. 基础算法（矩阵快速幂、构造算法）

### 3.2.3 搜索算法

1. 字符串哈希的原理与使用
2. 最小表示法的原理和使用

### 3.2.4 字符串算法

1. DFS 的剪枝与迭代加深
2. BFS 的双端队列和优先队列优化

### 3.2.5 动态规划

1. 状态压缩动态规划
2. 树上动态规划问题

### 3.2.6 图论算法

1. 最小生成树、次小生成树的原理与使用
2. 图上 DFS 序的性质与使用
3. 图上最短路算法——SPFA
4. Floyd 算法实现传递闭包
5. 欧拉路径与欧拉回路问题

### 3.2.7 数据结构

1. 特殊队列（双端队列、单调队列、优先队列）和特殊栈（双端栈）的原理与使用
2. 特殊数据结构 ST 表和并查集的原理与使用

### 3.2.8 基础数学

1. 基础数学（解析几何与立体几何）
2. 线性代数的基本概念、运算与转换
3. 初等数论（同余问题、概率与期望、逆元、拓展欧几里得算法）
4. 高中代数

### 3.3 BBCC CP VI (U18 高级) 知识点

#### 3.3.1 语言特性

1. (C++)STL 库(set、multiset、deque、priority\_queue、map、multimap、pair)的原理和使用
2. (Python)基本了解 Python 计算生态中常用的第三方库，包括但不限于常用于网络爬虫、计算科学、机器学习中的 numpy, scrapy, scikit-learn 等

#### 3.3.2 基础算法

1. 排序算法（桶排序、堆排序、树形选择排序）

#### 3.3.3 搜索算法

1. 优化搜索方法（记忆化搜索、双向搜索）
2. 启发式搜索

#### 3.3.4 字符串算法

1. KMP 算法的原理与使用

#### 3.3.5 动态规划

1. 动态规划的一系列常用优化（前缀和优化、单调队列优化、数据结构优化等）

#### 3.3.6 图论算法

1. 图上点分治和边分治算法
2. 二分图匹配算法
3. 双连通分量相关算法（点双连通、边双连通、割点、桥、强连通分量、缩点）
4. 树上倍增算法的原理与应用

#### 3.3.7 数据结构

1. 特殊树（二叉堆、树状数组、线段树、字典树、查找树、笛卡尔树、基环树）的原理与使用
2. 特殊图（二分图、欧拉图、强连通图）的原理与使用

### 3.3.8 基础数学

1. 重要定理（欧拉定理、费马小定理、威尔逊定理、裴蜀定理、中国剩余定理）的掌握和使用
2. 组合数学的使用（包括二项式定理、Lucas 定理、鸽巢原理、容斥原理、康托展开、卡特兰数

附表 1-1 BBCC U12 组 (BBCC CP I/II) 代码编程学习指南

BBCC/U12	知识点概述	知识点学时		知识点刷题 / 耗时
			合计	
初级	C++语言概述： 1. C++语言的基本符号 2. C++语言的词汇（关键字、标识符、常量、运算符、标点符号等） 3. C++程序的基本框架 4. Dev C++ 5.9.2 或 Visual C++ 2010 版集成开发环境的编辑、编译、运行	6	43	50 道左右 /70 小时
	C++数据类型、表达式和基本运算： 1. C++基本数据类型（整型、实型、字符型、布尔型）的类型描述、数据范围等 2. C++的常量定义及引用 3. 变量的定义与使用方法（变量的定义及初始化，全局变量，局部变量、变量的自增和自减运算） 4. C++运算符的种类、运算优先级和结合性	5		
	C++的基本语句： 1. 赋值语句、表达式语句、复合语句、输入、输出语句和空语句等 2. if、switch 语句实现分支结构 3. for、while、do...while 语句实现循环结构 4. continue、break 和 return 语句的运用	10		
	数组： 1. 一维数组的定义、初始化和访问 2. 多维数组的定义、初始化和访问 3. 字符串与字符数组的定义和使用	10		
	函数的有关使用： 1. 函数的定义方法和调用方法。 2. 函数的类型和返回值。 3. 形式参数与实际参数，参数值的传递	8		
	基础数学：算术运算（加、减、乘、除、整除、取余） 数制（进制概念、十进制、二进制、八进制、十六进制等概念及其相互转换）	4		

高级	Dev C++ 5.9.2 或 Visual C++ 2010 版集成开发环境的编辑、编译、运行、程序调试	1	52	60 道左右 /80 小时
	C++数据类型、表达式和基本运算： C++表达式类型及求值规则（赋值运算，算术运算符和算术表达式，关系运算符和关系表达式，逻辑运算符和逻辑表达式，条件运算，指针运算，逗号表达式）	3		
	C++的基本语句： 1. 分支语句的嵌套构造及其运用 2. 二层循环的构造及其运用 3. 多层循环的构造及其运用	6		
	数组： 常用字符串函数（长度、查找、比较、拷贝、连接、插入、截取、替换等）的使用	3		
	函数的有关使用： 1. 变量的作用域和生存周期 2. 递归函数	4		
	基础算法： 1. 时空复杂度分析（复杂度的概念、时间复杂度估算、空间复杂度估算） 2. 基础算法（模拟、枚举、贪心、递归、前缀和、分治与倍增、二分） 3. 排序（排序概念、冒泡排序、简单插入排序、简单选择排序、sort 函数）	20		
	简单数据结构： 1. 顺序表的概念及使用 2. 栈的概念及使用 3. 队列的概念及使用	10		
	基础数学： 基础数论（质数、合数、因数、倍数概念，质数的判断、辗转相除法算法及其应用）	5		

附表 1-2 BBCC U15 组(BBCC CP III/ IV)代码编程学习指南

BBCC/U15	知识点概述	知识点学时		知识点刷题/耗时
			合计	
初级	C++的数据类型和模板： 1. 结构体：结构体的声明与使用，运算符重载，结构体数组 2. 指针：指针运算符(& 和 *)的含义，指针的定义和使用指针与数组，指针与字符数组，结构体指针 3. 文件：文本文件的读写，文件的重定向操作	5	39	60 道左右/90 小时
	基础算法： 1. 高精度的表示方法与基本运算（加、减、乘和高精度与单精度的取余除法） 2. 基础算法（差分约束系统、尺取法与双指针、离散化思想） 3. 排序算法（计数排序、归并排序、快速排序）	10		
	图论算法：简单图上的搜索（DFS 与 BFS）	6		
	数据结构： 1. 链表：单向链表，双向链表，循环链表的构造与使用 2. 树的概念与表示方式（父亲表示法和孩子表示法） 3. 图的基本概念与存储方式（邻接表与邻接矩阵）	10		
	动态规划： 1. 动态规划的基本思想和基本原理 2. 常见线型动态规划问题，如求最长上升子序列、最长公共子序列等	6		
	基础数学：初中代数运算	2		
	C++的数据类型和模板：STL 库：vector、stack、queue、list 等模板的应用	5		
基础算法：简单分治、简单倍增、快速幂	5			
图论算法： 1. 二叉树的概念与存储方式 2. 二叉树的遍历（先序、中序、后序）	6			
动态规划： 1. 常见集合类型动态规划问题，如背包问题等 2. 常见划分类型规划问题，如石子合并等	6			
图论算法： 1. 两种最短路算法——Floyd 算法与 Dijkstra 算法 2. 有向无环图的拓扑排序	9			
数据结构：哈夫曼树的概念与应用	3			
基础数学：基础数论（素数筛、排列组合）	3			

附表 1-3 BBCC U18 组 (BBCC CP V/VI) 学习指南

BBCC/U18	知识点概述	知识点学时		知识点刷题/耗时
			合计	
初级	C++语言: C++中类(class)的声明与使用	1	74	70 道左右/120 小时
	字符串算法: 1. 字符串哈希的原理与使用 2. 最小表示法的原理和使用	6		
	搜索算法: 1. DFS 的剪枝与迭代加深 2. BFS 的双端队列和优先队列优化	6		
	图论算法: 1. 最小生成树、次小生成树的原理与使用 2. 图上 DFS 序的性质与使用 3. 图上最短路算法——SPFA 4. Floyd 算法实现传递闭包 5. 欧拉路径与欧拉回路问题	25		
	基础算法: 矩阵快速幂、构造算法	5		
	动态规划: 状态压缩动态规划, 树上动态规划问题	6		
	数据结构: 1. 特殊队列(双端队列、单调队列、优先队列)和特殊栈(双端栈)的原理与使用。 2. 特殊数据结构 ST 表和并查集的原理与使用	10		
	基础数学: 1. 基础数学(解析几何与立体几何) 2. 线性代数的基本概念、运算与转换 3. 初等数论(同余问题、概率与期望、逆元、拓展欧几里得算法) 4. 高中代数	15		
高级	C++语言中 STL 库(set、multiset、deque、priority_queue、map、multimap、pair)的原理和使用	3	109	80 道左右/140 小时
	基础算法: 排序算法(桶排序、堆排序、树形选择排序)	3		
	数据结构: 1. 特殊树(二叉堆、树状数组、线段树、字典树、查找树、笛卡尔树、基环树)的原理与使用。 2. 特殊图(二分图、欧拉图、强连通图)的原理与使用	30		
	字符串算法: KMP 算法的原理与使用	3		

<p>图论算法：</p> <ol style="list-style-type: none"> <li>1. 图上点分治和边分治算法</li> <li>2. 二分图匹配算法</li> <li>3. 双连通分量相关算法（点双连通、边双连通、割点、桥、强连通分量、缩点）</li> <li>4. 树上倍增算法的原理与应用</li> </ol>	25		
<p>搜索算法：启发式搜索，优化搜索方法（记忆化搜索、双向搜索）</p>	5		
<p>基础数学：</p> <ol style="list-style-type: none"> <li>1. 重要定理（欧拉定理、费马小定理、威尔逊定理、裴蜀定理、中国剩余定理）的掌握和使用</li> <li>2. 组合数学的使用（包括二项式定理、Lucas定理、鸽巢原理、容斥原理、康托展开、卡特兰数）。</li> </ol>	30		
<p>动态规划：一系列常用优化（前缀和优化、单调队列优化、数据结构优化等）</p>	10		

附表 2 BBCC 等级对照表（代码编程部分）

BBCC 认证等级（代码编程部分）与其他赛事对照表						
BBCC 组别	U12		U15		U18	
BBCC 等级	BBCC CP I	BBCC CP II	BBCC CP III	BBCC CP IV	BBCC CP V	BBCC CP VI
USACO	青铜组	青铜组-白银组		白银组	黄金组	黄金组-铂金组
NOIP	非高中选手只有分数 没有评奖		三等奖	三等奖 -二等奖	二等奖- 一等奖	一等奖
CSP/J/S	CSP/J 三等奖	CSP/J 二等奖-三等奖		CSP/J 二等奖	CSP/S 二等奖	CSP/S 二等奖- 一等奖

注：根据 BBCC 各个组别考察的知识点范围、内容、难度、学习时长等指标，对标当今国内及国际权威信息学竞赛。对标赛事简介如下：

1. CSP/J、CSP/S：中国计算机学会 (CCF) 组织的计算机软件能力非专业认证，J 组-入门组、S 组-提高组，从低到高分设三等奖到一等奖，类似 NOIP 的资格赛。
2. NOIP：中国计算机学会 (CCF) 组织的全国青少年信息学奥林匹克联赛，从低到高分设三等奖到一等奖。
3. USACO：美国计算机奥林匹克竞赛。难度等级从下往上共四个等级：青铜级、白银级、黄金级、铂金。

## 附录 1 代码编程知识点索引

基本数据类型	BBCC CP I	线型动态规划	BBCC CP III
常量与变量	BBCC CP I	划分类型动态规划	BBCC CP IV
运算符	BBCC CP I	集合类型动态规划	BBCC CP IV
类型转换	BBCC CP I	快速幂	BBCC CP IV
分支结构	BBCC CP I/ II	二叉树	BBCC CP IV
循环结构	BBCC CP I/ II	基础数论	BBCC CP IV
异常处理	BBCC CP I	Floyd 与 Dijkstra 算法	BBCC CP IV
列表和元组	BBCC CP I	有向无环图的拓扑排序	BBCC CP IV
函数参数	BBCC CP I/II	哈夫曼树	BBCC CP IV
作用域与生存周期	BBCC CP II	矩阵快速幂	BBCC CP V
递归函数	BBCC CP II	DFS 与 BFS 的优化	BBCC CP V
字典	BBCC CP I	状态压缩动态规划	BBCC CP V
数制	BBCC CP I	最小生成树	BBCC CP V
时空复杂度	BBCC CP II	次小生成树	BBCC CP V
模拟法	BBCC CP II	SPFA 算法	BBCC CP V
枚举法	BBCC CP II	传递闭包	BBCC CP V
贪心法	BBCC CP II	欧拉路径与欧拉回路	BBCC CP V
递归法	BBCC CP II	双端队列	BBCC CP V
前缀和	BBCC CP II	单调队列	BBCC CP V
分治与倍增	BBCC CP II	优先队列	BBCC CP V
二分法	BBCC CP II	双端栈	BBCC CP V
类的继承	BBCC CP II	ST 表	BBCC CP V
多态与方法重载	BBCC CP II	并查集	BBCC CP V
计算生态	BBCC CP II/VI	字符串哈希	BBCC CP V
结构体	BBCC CP III	最小表示法	BBCC CP V
指针运算符	BBCC CP III	桶排序	BBCC CP VI
引用	BBCC CP III	堆排序	BBCC CP VI
高精度运算	BBCC CP III	树形选择排序	BBCC CP VI
差分约束系统	BBCC CP III	记忆化搜索	BBCC CP VI
尺取法与双指针	BBCC CP III	双向搜索	BBCC CP VI
离散化思想	BBCC CP III	树上动态规划	BBCC CP VI
基数排序	BBCC CP III	图上点分治和边分治算法	BBCC CP VI
归并排序	BBCC CP III	二分图匹配算法	BBCC CP VI
快速排序	BBCC CP III	双连通分量相关算法	BBCC CP VI
DFS 与 BFS	BBCC CP III/V	树上倍增算法	BBCC CP VI
链表	BBCC CP III	KMP 算法	BBCC CP VI
图存储方式	BBCC CP III	启发式搜索	BBCC CP VI
动态规划基本思想	BBCC CP III	动态规划常用优化	BBCC CP VI

## 联系我们

深圳市计算机学会（SZCCF）致力于与会员共享、共建、共赢。

学会有多个不同领域专业划分的专委会；立足计算机学科助力培养青少年科技后备人才的青少年工作委员会；培养9岁到19岁青少年算法和编程能力的大湾区青少年信息学编程认证(BBCC)；为促进计算机科学与技术的创新和进步而设立的深圳市计算机学会奖（SZCCF奖）；全年数十场会员交流论坛、会议。

加入SZCCF将获得更聚焦的行业人脉，更精准的专业助力，更优质的职业发展机遇。

扫码加入SZCCF



联系我们：

张老师，联系电话：18902844857

微信号：SZCCF2020（张老师）

icnlab（刘老师）

联系邮箱：[contact@szccf.org.cn](mailto:contact@szccf.org.cn)

深圳市计算机学会 (SZCCF)