这不是总结！这不是总结！这不是总结！重要的事情说三遍！

1. 计算过程与计算思维：十种理解

理解 1：**自动执行**。计算机能够自动执行由离散步骤组成的计算过程。
理解 2：**正确性**。计算机求解问题的正确性往往可以精确地定义并分析。
理解 3：**通用性**。计算机能够求解任意可计算问题。
理解 4：**构造性**。人们能够构造出聪明的方法让计算机有效地解决问题。
理解 5：**复杂度**。这些聪明的方法（算法）具备时间/空间复杂度。
理解 6：**连通性**。很多问题涉及用户/数据/算法的连接体，而非单体。
理解 7：**协议栈**。连接体的节点之间通过协议栈通信交互。
理解 8：**抽象化**。少数精心构造的计算抽象可产生万千应用系统。
理解 9：**模块化**。多个模块有规律地组合成为计算系统。
理解 10：**无缝衔接**。计算过程在计算系统中流畅地执行。

2. 计算思维的五个主要目标：自动、通用、算法、联网、抽象。

- **自动**。这个目标对应第一个理解：离散步骤自动执行。计算机科学首先解决的科技难题是将描述物理世界和数学世界的各种公式和方程离散化、数字化，并将连续时间变成离散的步骤，让求解这些公式和方程的计算过程能够自动执行，而不是像使用算盘那样，每一步都需要人工操作。自动执行有助于大幅度提升计算速度，降低计算成本。
- **通用**。这个目标对应第二、三个理解：正确性与通用性。在解决了某些数学方程的计算过程自动执行问题之后，人们很自然地猜测：什么样的计算过程能够自动执行呢？是不是所有的计算过程都能被计算机自动执行？是不是所有求解数学方程的计算过程都能被计算机自动执行？存不存在某一种计算机，它能够用于执行所有计算过程、解决所有计算问题？通用性难题的一个重要里程碑发生在 1936 年，图灵从理论上提出了一种通用计算机，能够求解任意可计算问题。而要做到这一点，必须精确地定义并分析计算机求解问题的正确性，包括定义什么是可计算问题、什么是不可计算问题。
- **算法**。这个目标对应第四、五个理解：构造性与复杂度。知道某个问题可计算还远远不够。一个问题可能通过无穷多种计算过程得到解决。我们需要找到或构造出有效的方法（称为算法），能够花费较短的计算时间、使用较少的计算资源，通过执行比较聪明的特定计算过程来解决问题。
- **联网**。这个目标对应第六、七个理解：连通性与协议栈。人们很早就注意到了，很多问题只用单点执行的算法不能解决。比如，互联网搜索问题（即搜索数以百万计的互联网网站的信息，找到相关答案），单点算法就不能解决。我们需要将多点的数据和算法连接起来、组成网络，才能解决这些问题。
- **抽象**。这个目标对应第八、九、十个理解：抽象化、模块化、无缝衔接。抽象（abstraction）贯穿计算机科学的发展史，是计算机科学最重要的研究目标和研究对象，同时又是较难掌握的计算机科学的根本方法。计算过程都是在计算系统中执行的。计算机科学并不是针对每一个问题、每一个计算过程设计一套计算系统，而是归纳出少数精心构造的计算抽象，有规律地组合起来，产生万千应用系统。也就是说，计算机科学讲究尽量用一个（或一套）通用抽象支持众多具体应用需求。自动执行、通用、算法、网络都涉及抽象。

3. **计算思维理解之一：计算过程的有限步骤与自动执行**

计算过程刻画：

1. 一个计算过程是解决某个问题的有限个计算步骤的执行序列。
2. 计算过程有一个起始步骤（第一步骤）。
3. 每个步骤执行一个操作，然后进行到下一步骤。
   a) 每个操作可以是一个算术逻辑运算操作，也可以是一个访存操作、一个跳转操作、或是一个输入输出操作。它们都可被视为数字符号变换操作。
   b) 下一步骤可以是顺序下一步骤，也可以是跳转操作的目的步骤。
4. 当不存在下一步骤时，计算过程终止。

计算思维要点："精准地描述信息变换过程的操作序列，并有效地解决问题"是如何体现的？。

- 精准性在于：
  （1）精确地定义该问题涉及的数字符号、操作和步骤；
  （2）每个符号、操作、步骤要足够能行，即人用纸和笔在较短时间能够实现。
- 有效性体现于：
  除了输入输出操作之外，计算过程的所有步骤和操作能够在一个计算机中自动执行。解决"30 亿个数的求和问题"的计算过程共需要执行 90 亿余个步骤。假设 30 亿个数已经载入存储器，每个步骤费时 1 纳秒，则自动执行的求和计算仅需费时大约 90 亿纳秒=9 秒钟！

因此，计算机科学的一个里程碑是攻克了自动执行的难题。

4. 计算思维理解之二：计算过程的正确性

计算过程刻画：

1. 一个计算过程是解决某个问题的有限个计算步骤的执行序列。该计算过程的功能就是正确地解决这个问题。
2. 人们可用布尔逻辑描述计算过程的基本操作和步骤。
3. 人们也可以用布尔逻辑描述待解问题。
4. 计算过程的正确性体现为计算过程的结果在逻辑上符合问题的解，或从逻辑上显示解不存在。

计算思维要点："精准地描述信息变换过程的操作序列，并有效地解决问题"是如何体现的？。

- 精准性在于：
  用布尔逻辑精确地定义该问题涉及的数字符号、操作和步骤，以及问题的解。
- 有效性体现于：
  （1）对问题合理地建模，使得求解更加容易；
  （2）利用计算机自动执行特性实现逻辑推理自动化。

5. 计算思维理解之三：图灵机的通用性

计算过程刻画：

1. 任何可计算问题都能被描述为图灵机上的计算过程。
2. 人们可用图灵机描述计算过程。也就是说，计算过程的基本操作、步骤，以及计算步骤的执行序列，都可以被映射到图灵机上。

3. 存在图灵机不可计算的问题：即使在通用图灵机上，也找不到计算过程，能够给出该问题的解。

计算思维要点："精准地描述信息变换过程的操作序列，并有效地解决问题"是如何体现的？。

- 精准性在于：
  对于图灵可计算问题，用图灵机精确地定义该问题涉及的数字符号、操作和步骤，以及问题的解。
- 有效性体现于：
  （1）对问题合理地建模，使得求解更加容易；
  （2）利用图灵机的自动执行特性实现计算过程的自动化。

6. 计算思维理解之六：连通性

很多问题涉及用户/数据/算法的连接体，而非单体

计算过程刻画：

1. 一个计算过程是解决某个问题的有限个计算步骤的执行序列。
2. 计算过程的整体或一个步骤可能需要将连接体作为处理对象。
3. 计算过程的整体或一个步骤可能需要在连接体上执行。

计算思维要点："精准地描述信息变换过程的操作序列，并有效地解决问题"是如何体现的？

- 用名字空间和网络拓扑精准地描述连接体，即操作对象或执行系统。
- 在问题建模或解题过程中，不只使用单点做计算，而是采用连接体（即多个节点互联而成的网络）作为计算对象或计算系统。

7. 计算思维理解之七：协议栈

节点之间通过协议栈传递消息

计算过程刻画：

1. 计算过程可包含消息传递步骤。
2. 消息传递的核心操作是包交换，即通过"包头+包体"的消息包传递信息。

计算思维要点："精准地描述信息变换过程的操作序列，并有效地解决问题"是如何体现的？

- 精准地描述协议栈整体，以及每个协议的消息格式、层间接口、对等接口。
- 针对问题，确定应用协议层次，充分重用通用的互联网协议。

8. 计算思维理解之八：抽象化

少数精心构造的计算抽象可产生万千应用系统

计算过程刻画：

1. 一个计算过程是解决某个问题的有限个计算步骤的执行序列。
2. 每个计算步骤都是对数据抽象的操作，并受控制抽象约束。
3. 计算步骤有大有小。大的计算步骤由小的计算步骤组合而成，是一组小的计算步骤的抽象。

计算思维要点：计算系统思维方法的要点是通过抽象将部件组合成为系统，执行计算过程。

- 计算抽象既是计算机科学最重要的方法，也是最重要的产物。作为动名词的抽象也被称为抽象化，而抽象化的产物也称为抽象，两者对应的英文都是 abstraction。计算抽象主要包括数据抽象与控制抽象。

抽象化和抽象具备抽象三性质：

- 有限性：每个抽象仅仅考虑一个层次的有限的特有问题，忽略其他层次和其他问题。
- 精确性：每个抽象是一个语义精确、格式规范的计算概念。
- 普遍性：每个抽象都具有泛化能力。

9. 计算思维理解之九：模块化

多个模块有规律地组合成为计算系统；系统=模块的组合

计算过程刻画：

1. 一个计算过程是解决某个问题的有限个计算步骤的执行序列。
2. 一个计算过程或一个步骤可能需要多个模块共同执行。

计算思维要点：模块化是一类特殊的抽象化方法，其要点是理解如何从部件（即模块）组合系统，需要回答**系统架构三问题**：

- 系统是由哪些模块组成的？
- 系统是由这些模块如何组成的（模块之间如何接口）？
- 计算过程在系统中如何执行？

信息隐藏原理有助于理解并控制计算系统的复杂性。

10. 计算思维理解之十：无缝衔接

计算过程在计算系统中无缝流畅地执行。

计算过程刻画：

一个计算过程是有限个计算步骤的执行序列，两个相邻的步骤之间需要无缝过渡，没有缝隙和瓶颈，从一步骤到下一步骤自动流畅地执行。

计算思维要点：

理解无缝衔接需要理解四条原理：扬雄周期原理、波斯特尔鲁棒性原理、冯诺依曼穷举原理、阿姆达尔定律。前三者主要应对缝隙问题，后者主要应对瓶颈问题。它们合起来使得计算步骤可以级联起来，实现计算过程。

11. 什么是算法？

高德纳的算法定义：一个算法是一组有穷的规则，给出求解特定类型问题的运算序列，并具备下列五个特征：

（1）有穷性：一个算法在有限步骤之后必然要终止。

（2）确定性：一个算法的每个步骤都必须精确地（严格地和无歧义地）定义。

（3）输入：一个算法有零个或多个输入。

（4）输出：一个算法有一个或多个输出。

（5）能行性：一个算法的所有运算必须是充分基本的，原则上人们用笔和纸可以在有限时间内精确地完成它们。