

计算机程序设计

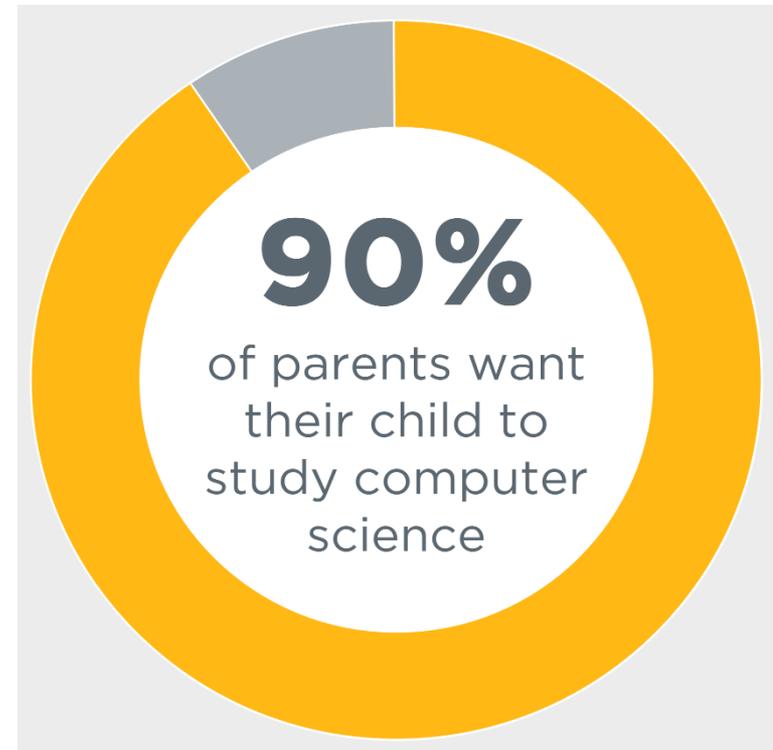
李昊

hao.li@xjtu.edu.cn

西安交通大学计算机学院

为什么学计算机？

为什么学计算机？



为什么学计算机?

1、关注度最高的本科专业TOP10对比

	2021年	2022年	2023年
1	计算机科学与技术	法学	计算机科学与技术
2	法学	计算机科学与技术	法学
3	临床医学	临床医学	临床医学
4	英语	汉语言文学	人工智能
5	汉语言文学	软件工程	电气工程及其自动化
6	金融学	英语	口腔医学
7	心理学	人工智能	汉语言文学
8	自动化	口腔医学	软件工程
9	土木工程	金融学	自动化
10	电气工程及其自动化	心理学	电子信息工程

头条@八维说

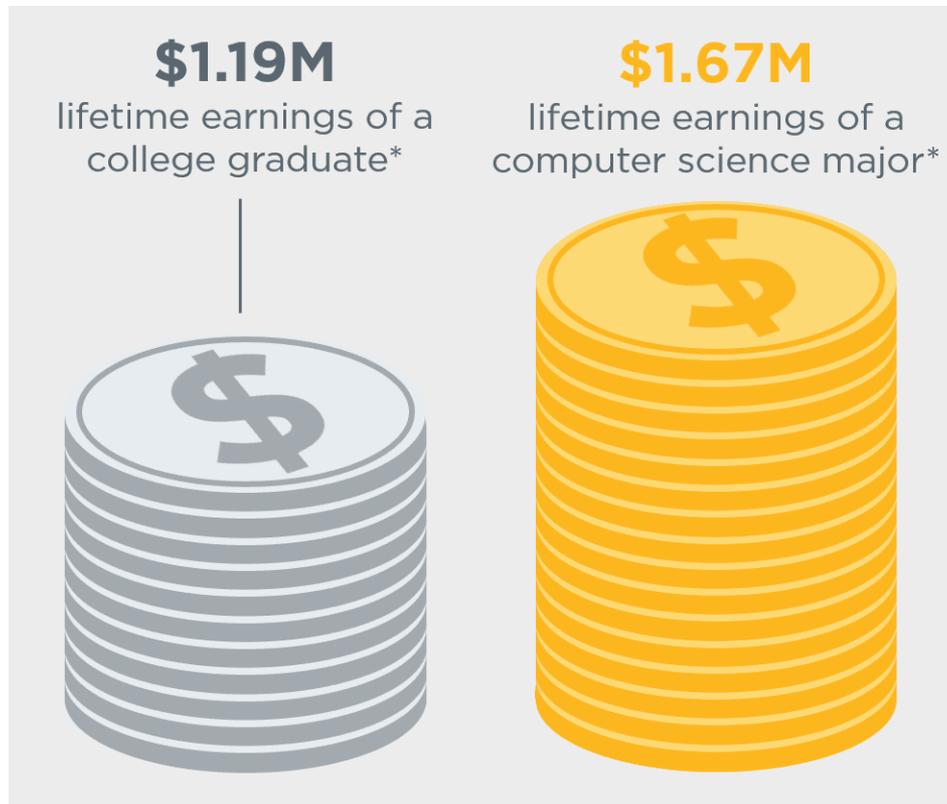
还在迷茫

张雪峰老师推荐的十大高薪专业

- 1、信息安全（连续七年薪资最高的专业，共上榜8次）
- 2、软件工程（唯一一个连续十年入围Top10的专业）
- 3、信息工程icon（上榜9次）
- 4、网络工程（上榜8次）
- 5、计算机（这些专业不仅是当前热门）
- 6、电子科学与技术（上榜5次）
- 7、物联网工程icon（上榜5次）
- 8、数字媒体技术（近五年薪资稳步上升）
- 9、法语（上榜5次）
- 10、信息管理与信息系统（十年来薪资涨

+ 关注

为什么学计算机？



计算机专业毕业生
薪水高于毕业生平
均值**40%**
(2023, 美国)

计算机学什么？

计算机学什么？

Computer Science Major



What my friends think I do.



What my mom thinks I do.



What society thinks I do.



What my professor thinks I do in class.

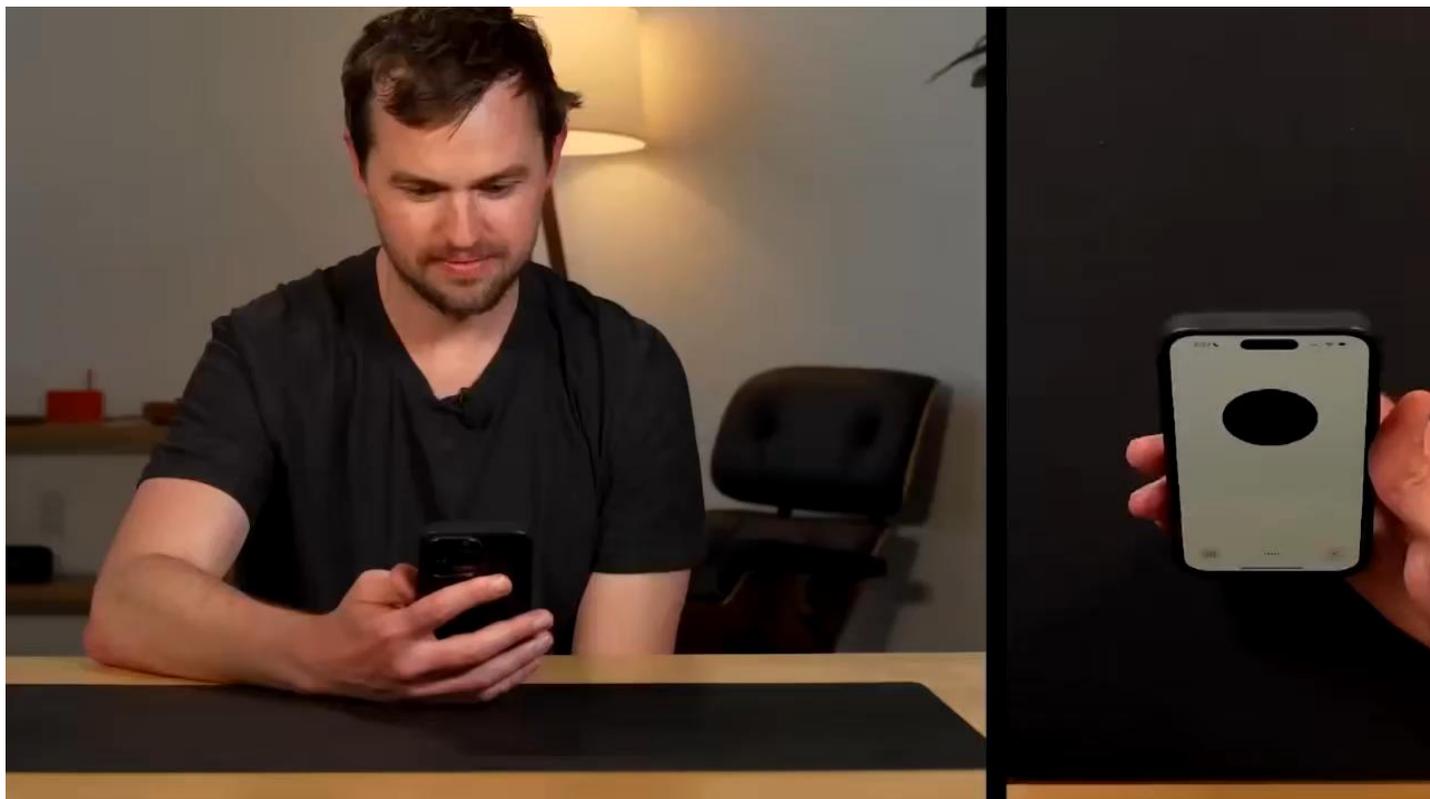


What I think I do.



What I actually do.

计算机学什么？



生成式人工智能

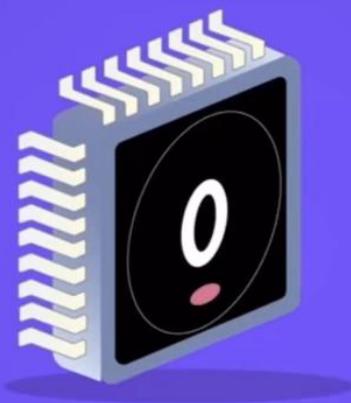
计算机学什么？



芯片设计与制造

计算机学什么？

史丹福机器人庞博士 bilibili

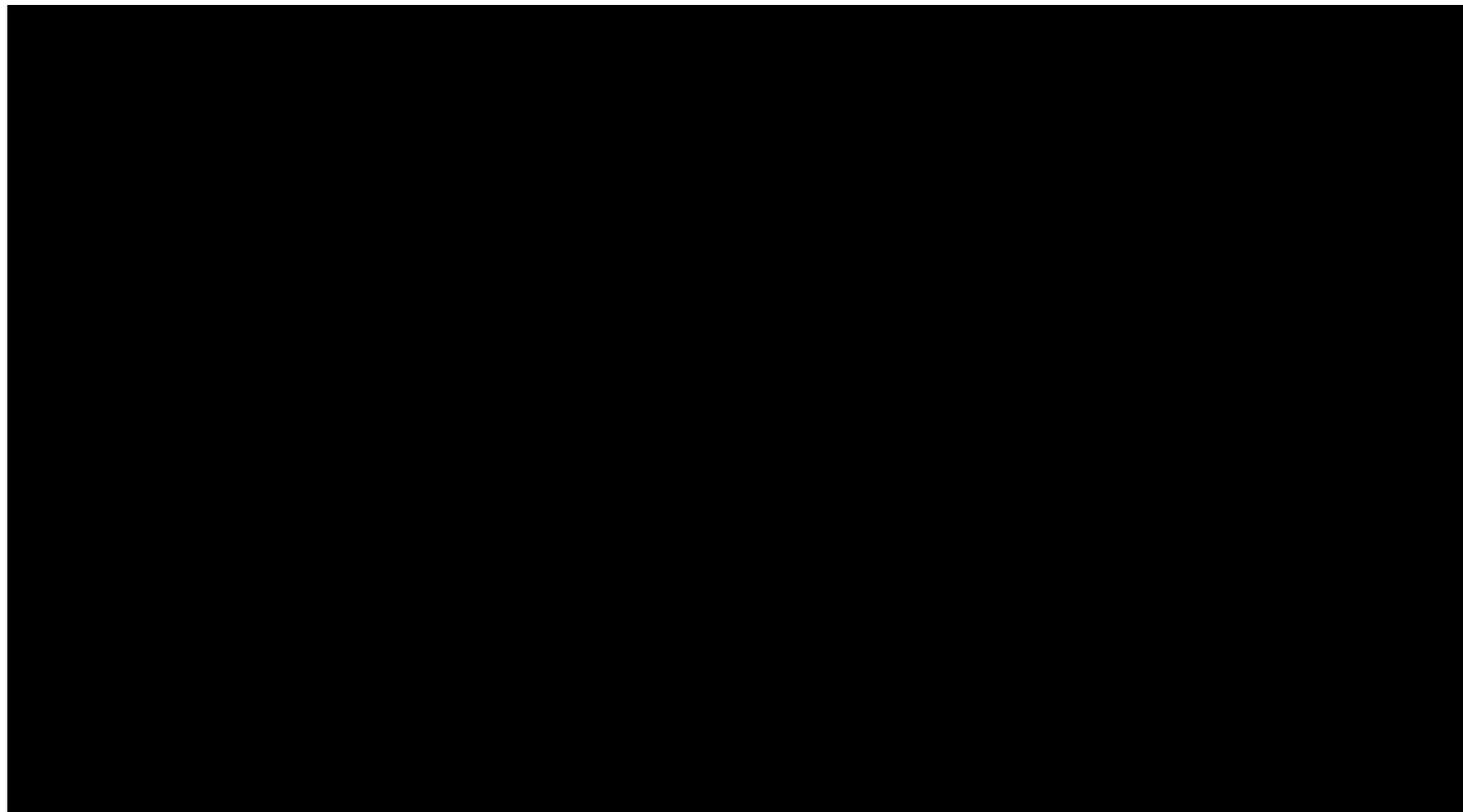


1

0 我们知道计算机的比特只有0和1

新型计算机体系结构

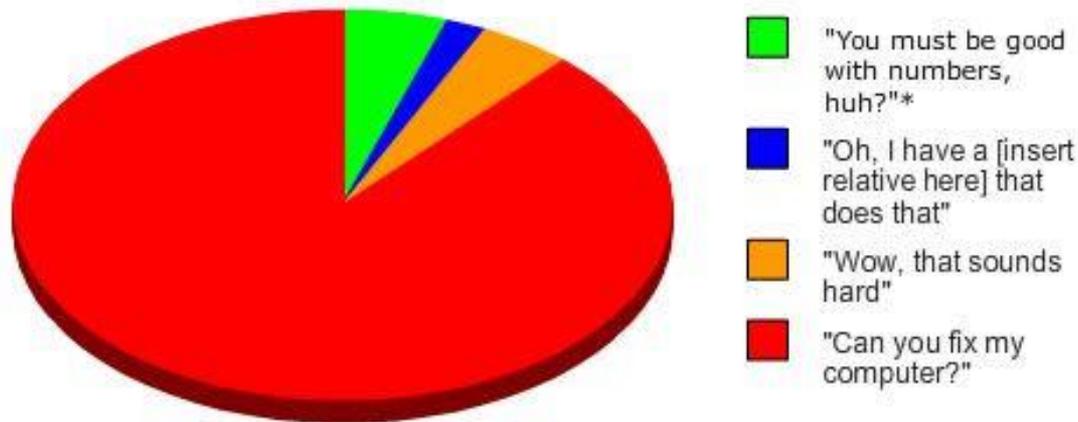
计算机学什么？



次世代游戏与影视CG

计算机学什么？

Other People's Responses to the fact that I am a Computer Science Major



*To which my typical response is "Yeah, mostly ones and zeroes"

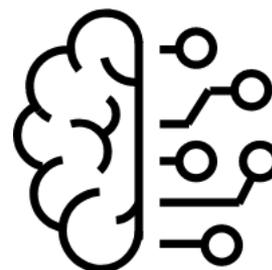
计算机学什么？



计算机科学与技术



物联网



人工智能



网络安全

计算机学什么？

编程！

计算机体系结构
(研究计算机本身)



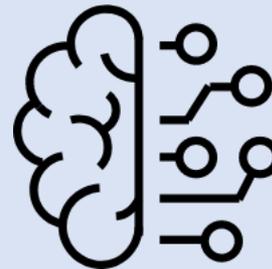
计算机科学与技术

计算机核心应用

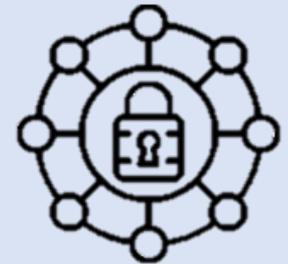
(研究如何利用计算机优化应用)



物联网



人工智能



网络安全

计算机程序设计

Programming Fundamentals

教学团队



李昊 副教授

hao.li@xjtu.edu.cn

兴庆校区 彭康楼234



唐亚哲 教授

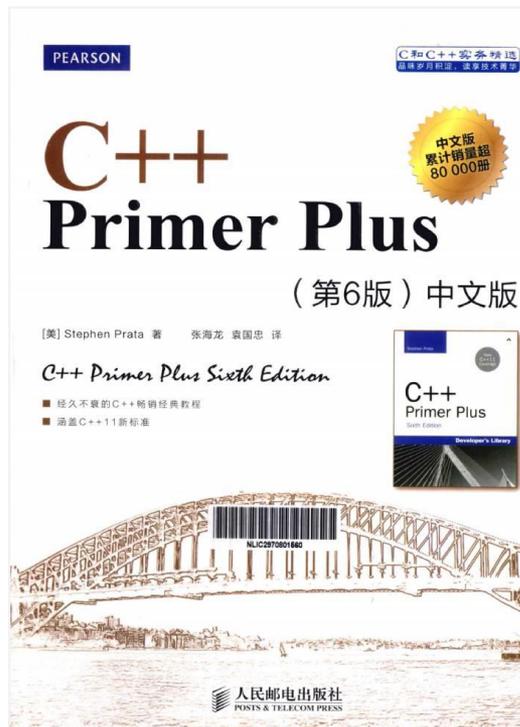
yztang@xjtu.edu.cn

兴庆校区 西一楼441

教材与参考书

核心教材

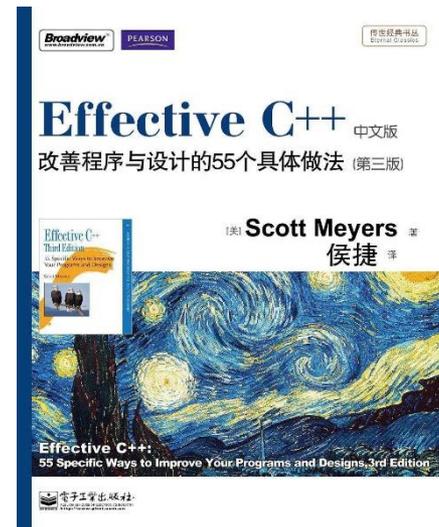
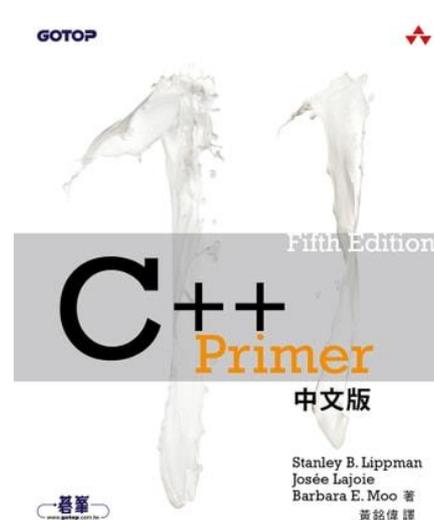
C++ Primer Plus 6th



参考材料

C++ Primer 5th

Effective C++ 3rd



教学规划 – 考核

平时成绩 **20%**

考勤、课堂测验

Auto-Grading系统对代码自动打分

Anti-Cheating系统自动检测代码抄袭

抄袭是**高压线**，一经核实双方对应测验分数为0

期末考试 **80%**

全机试，写代码

课程主题

编程 = 写作

编程：使用程序设计语言向计算机传递信息

写作：使用自然语言向读者传递信息

好的程序 = 好的文章

如何学好编程 = 如何学好写作

第一阶段：语法

单词+语法



苹果



Apple

关键字+语法

```
int main()  
{  
    cout << "Hello, world!" << endl;  
    return 0;  
}
```

第二阶段：语义

语法正确



苹果爱我

语义正确



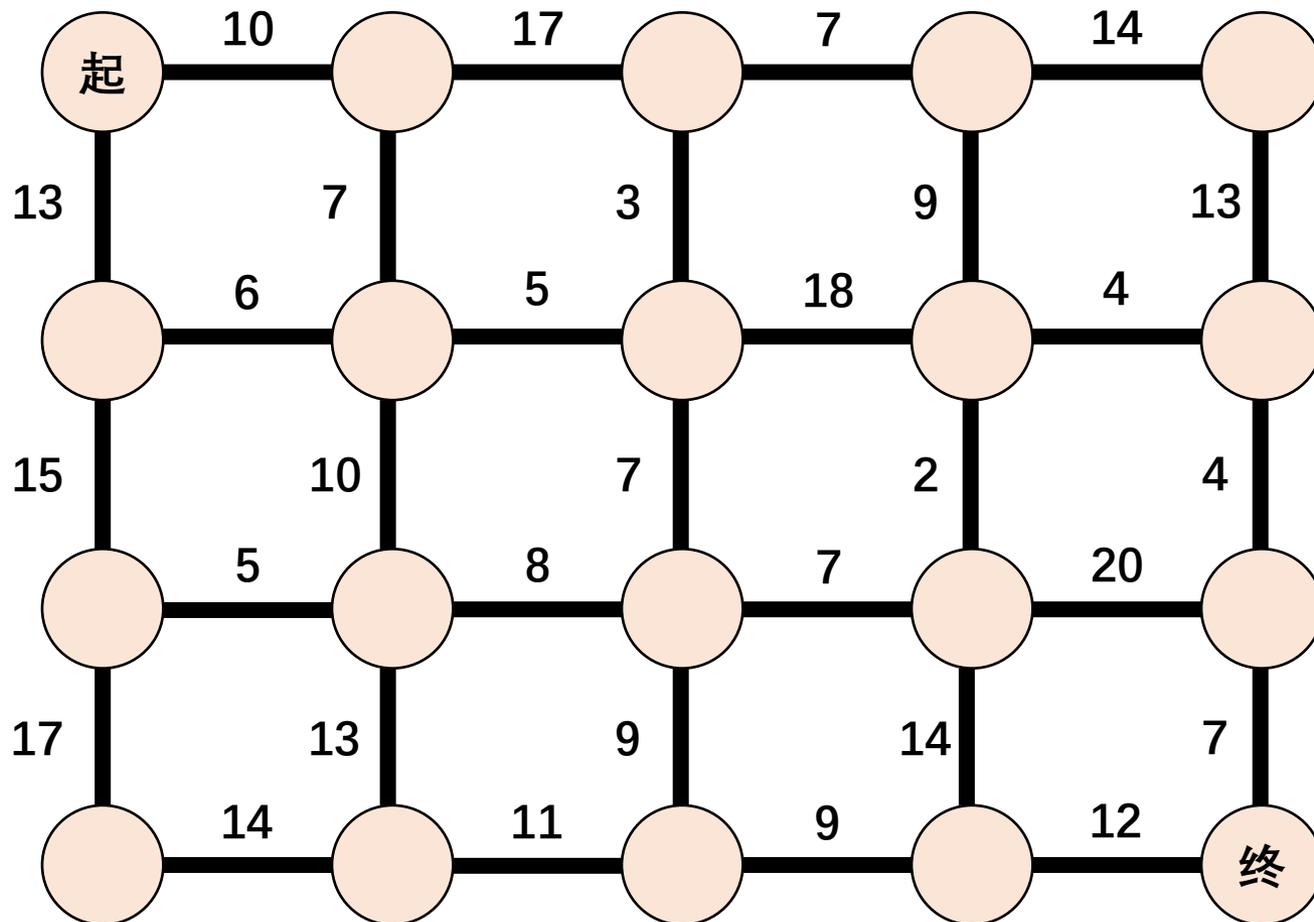
我爱苹果

语义高效/充分

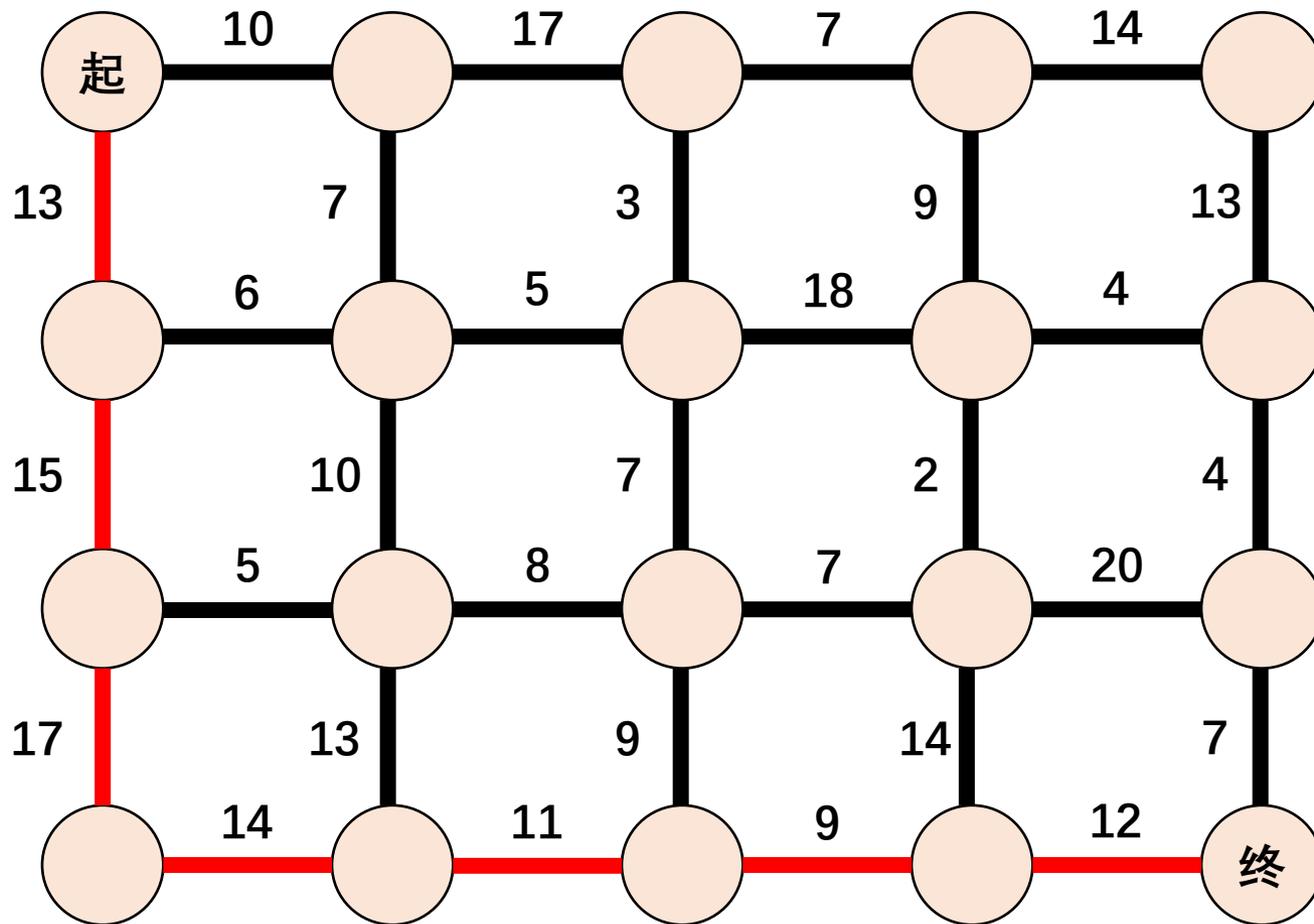


我爱那甜美多汁的
红苹果，如同爱那
初春的阳光。

寻找最短路径

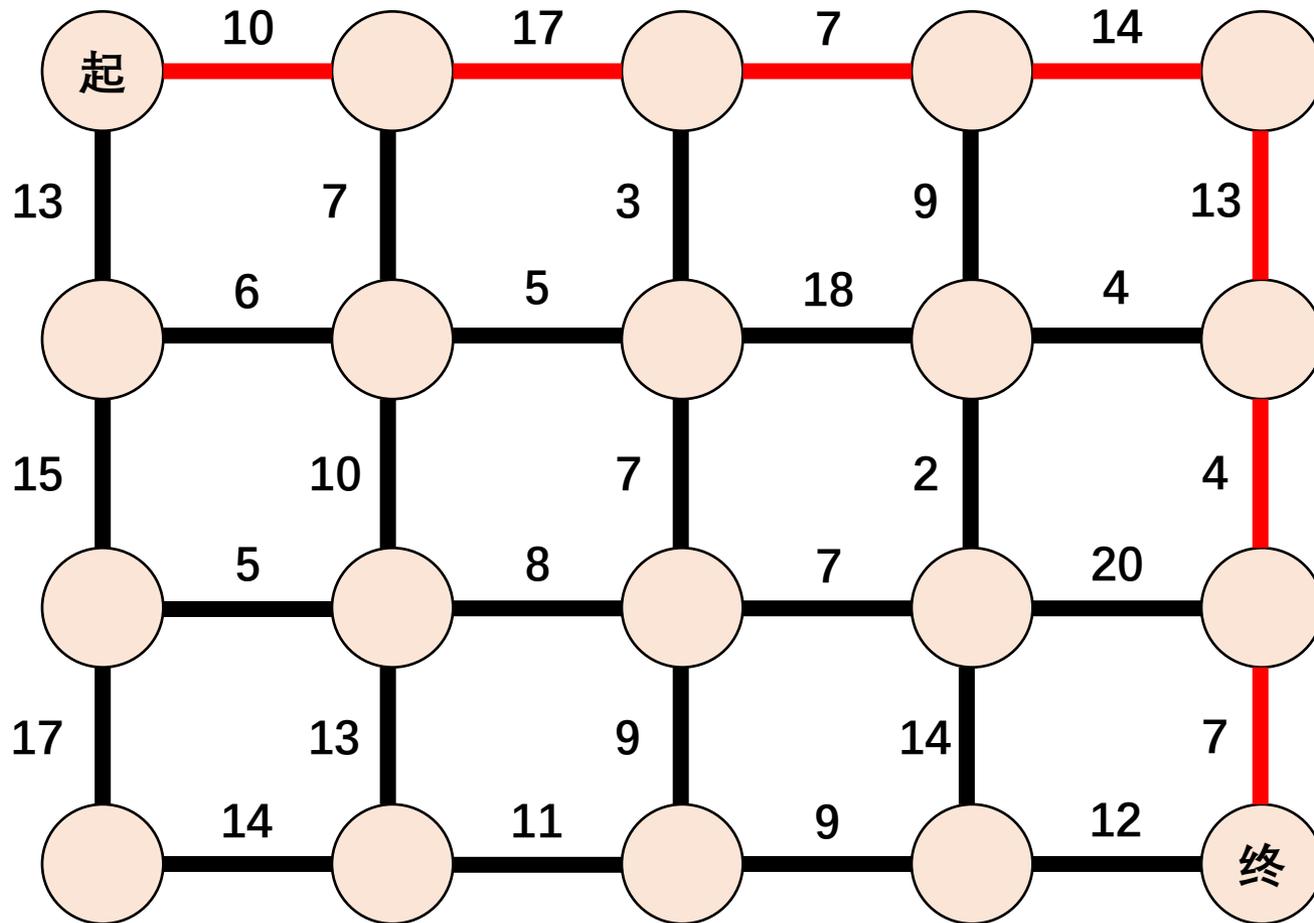


寻找最短路径



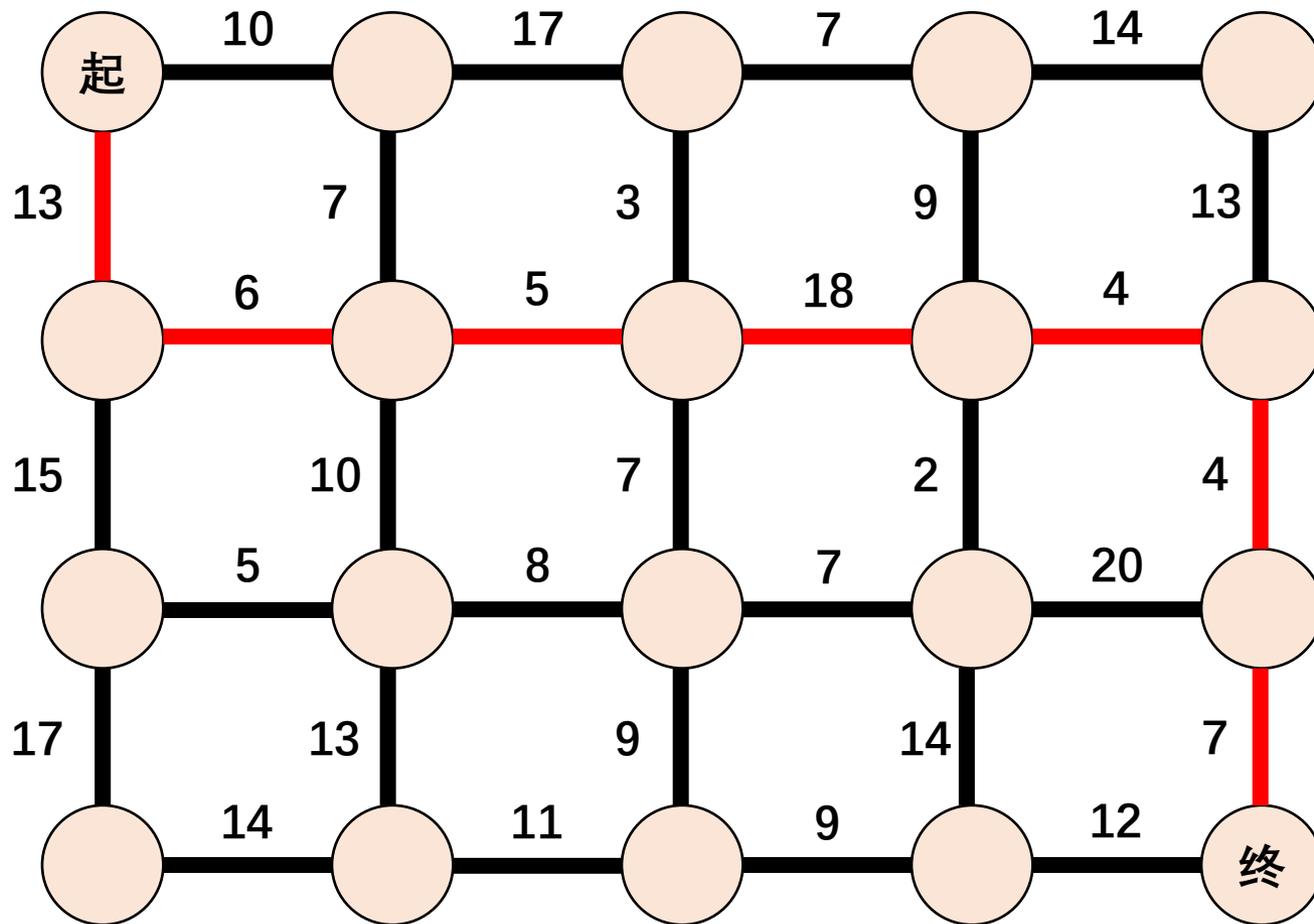
$$13 + 15 + 17 + 14 + 11 + 9 + 12 = 91$$

寻找最短路径

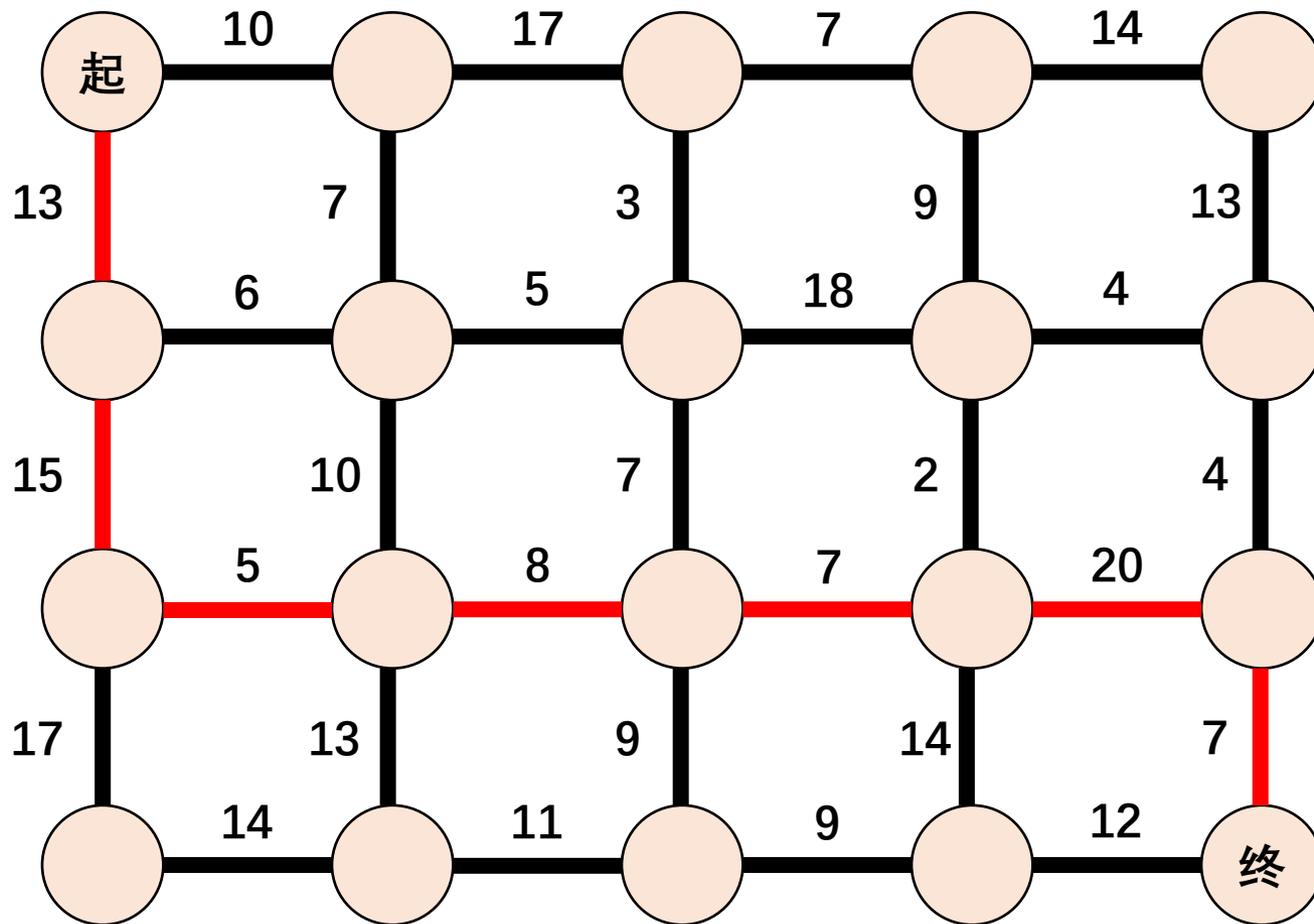


$$10 + 17 + 7 + 14 + 13 + 4 + 7 = 72$$

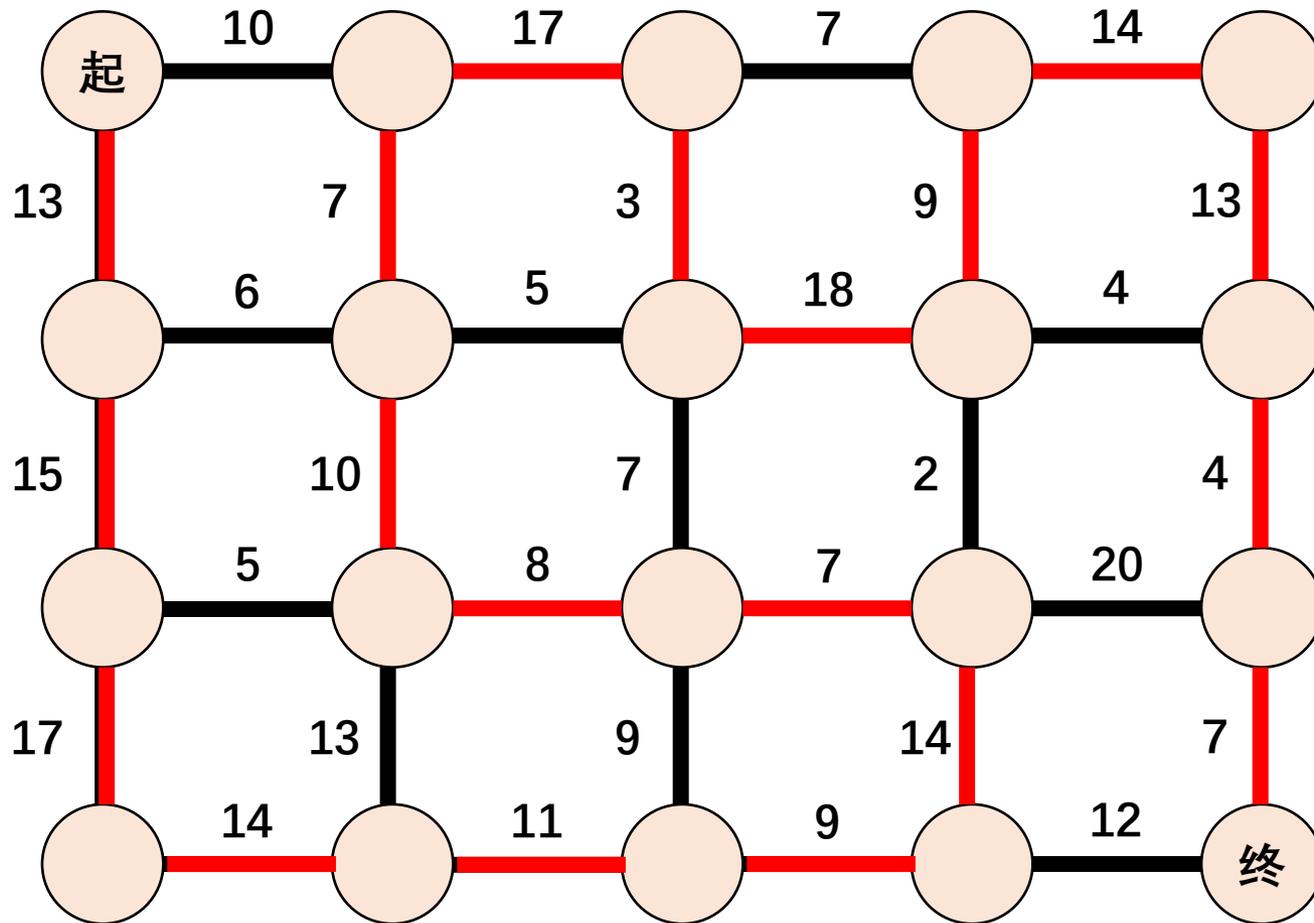
寻找最短路径



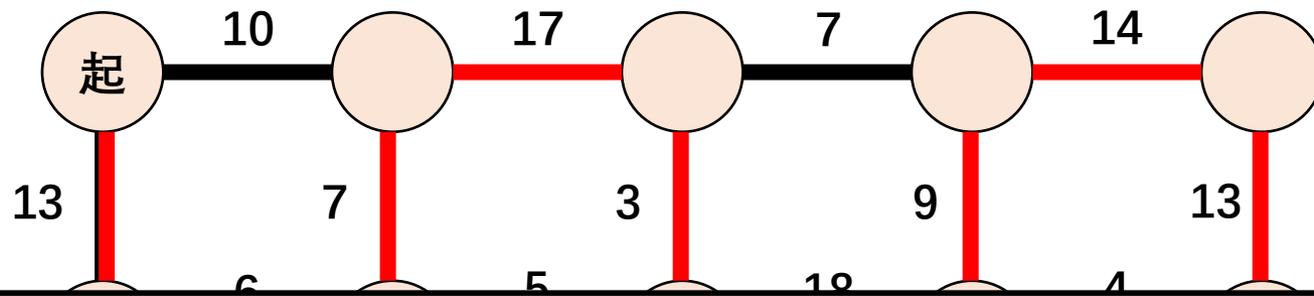
寻找最短路径



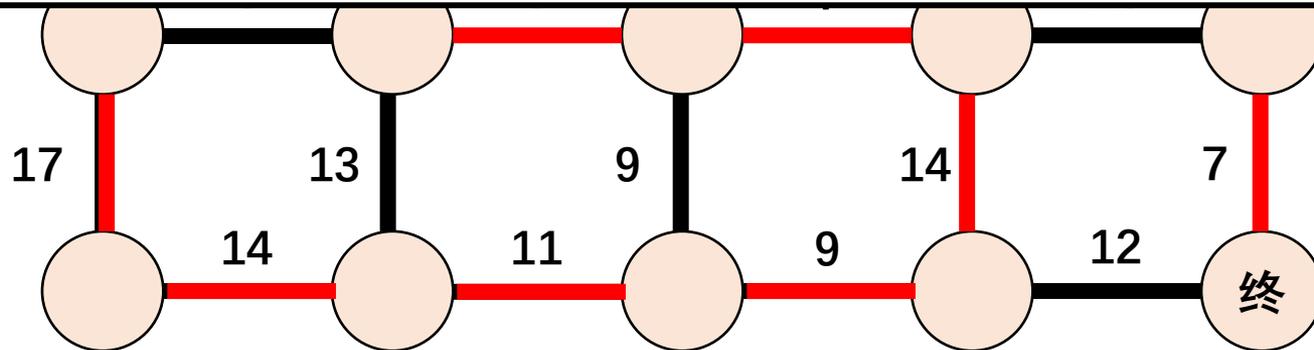
寻找最短路径



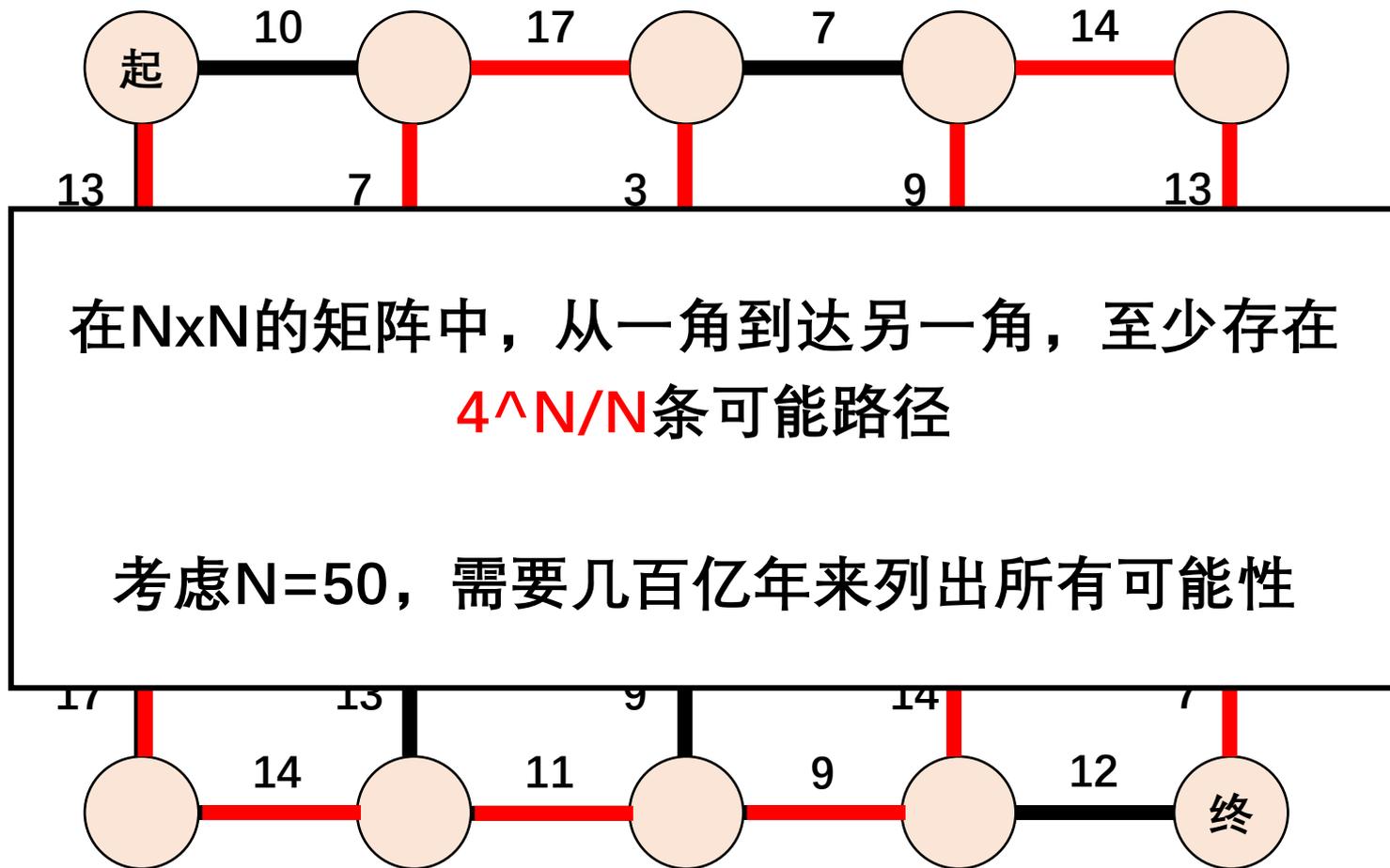
寻找最短路径



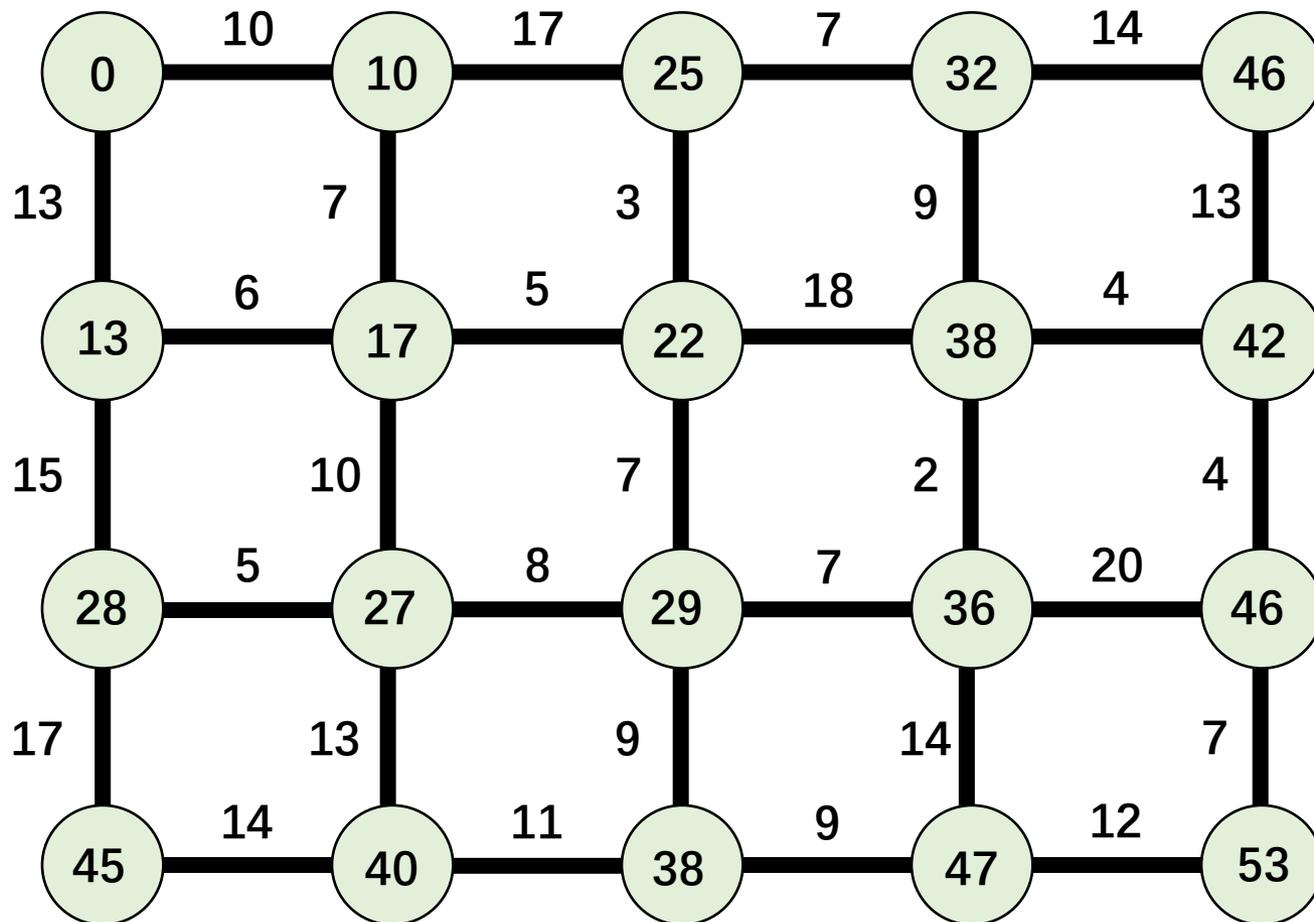
在 $N \times N$ 的矩阵中，从一角到达另一角，至少存在
 $4^{N/2}$ 条可能路径



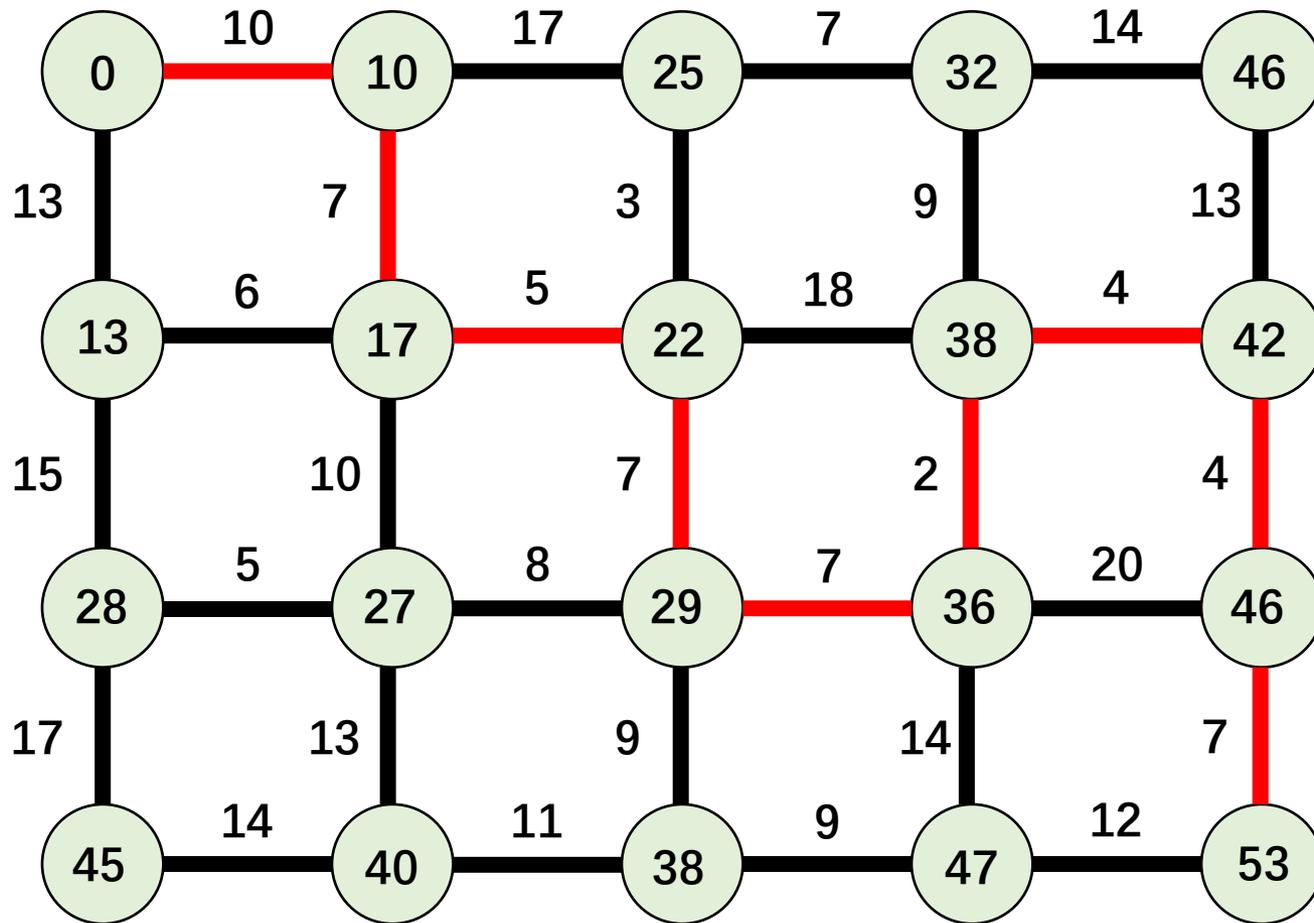
寻找最短路径



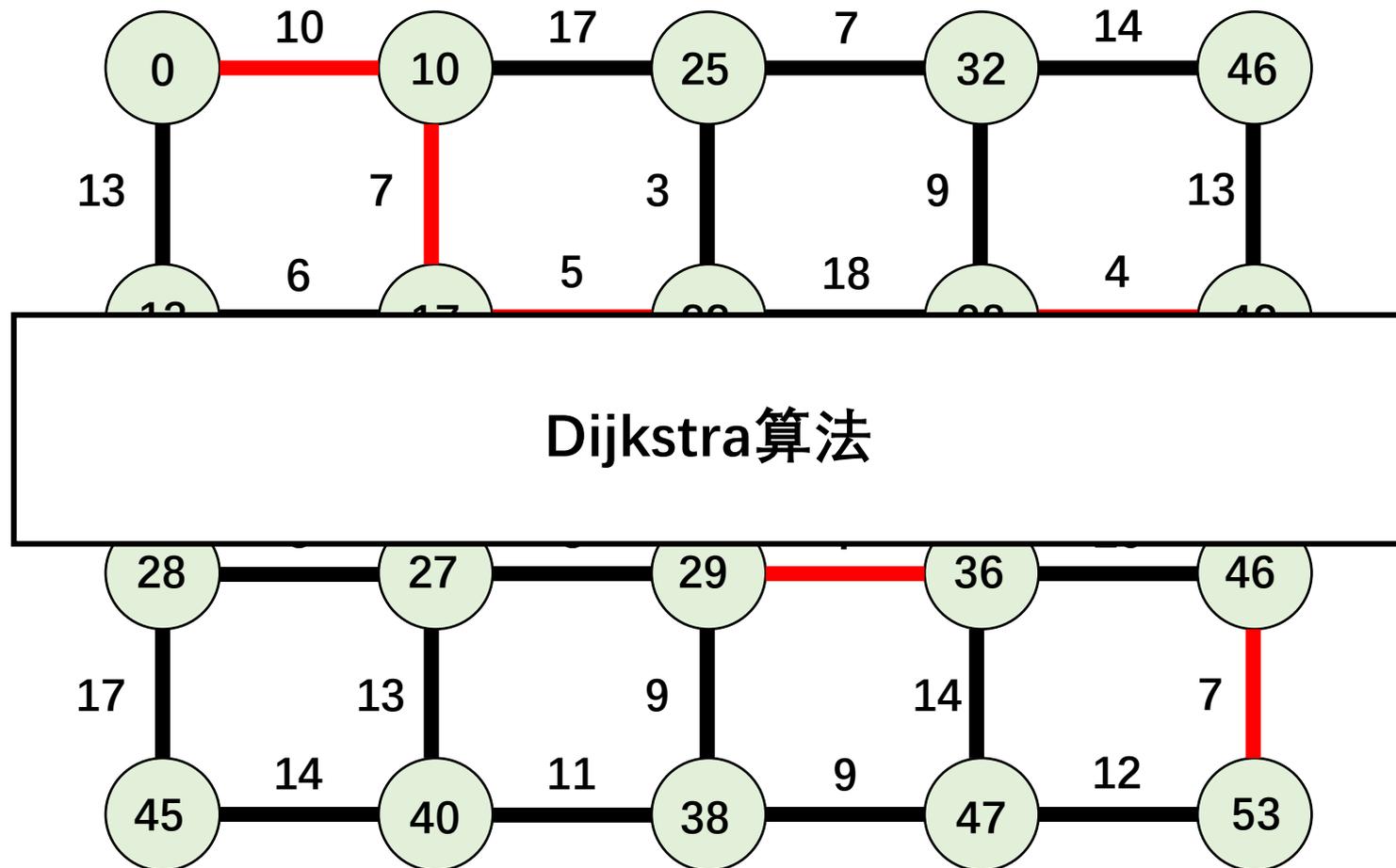
寻找最短路径



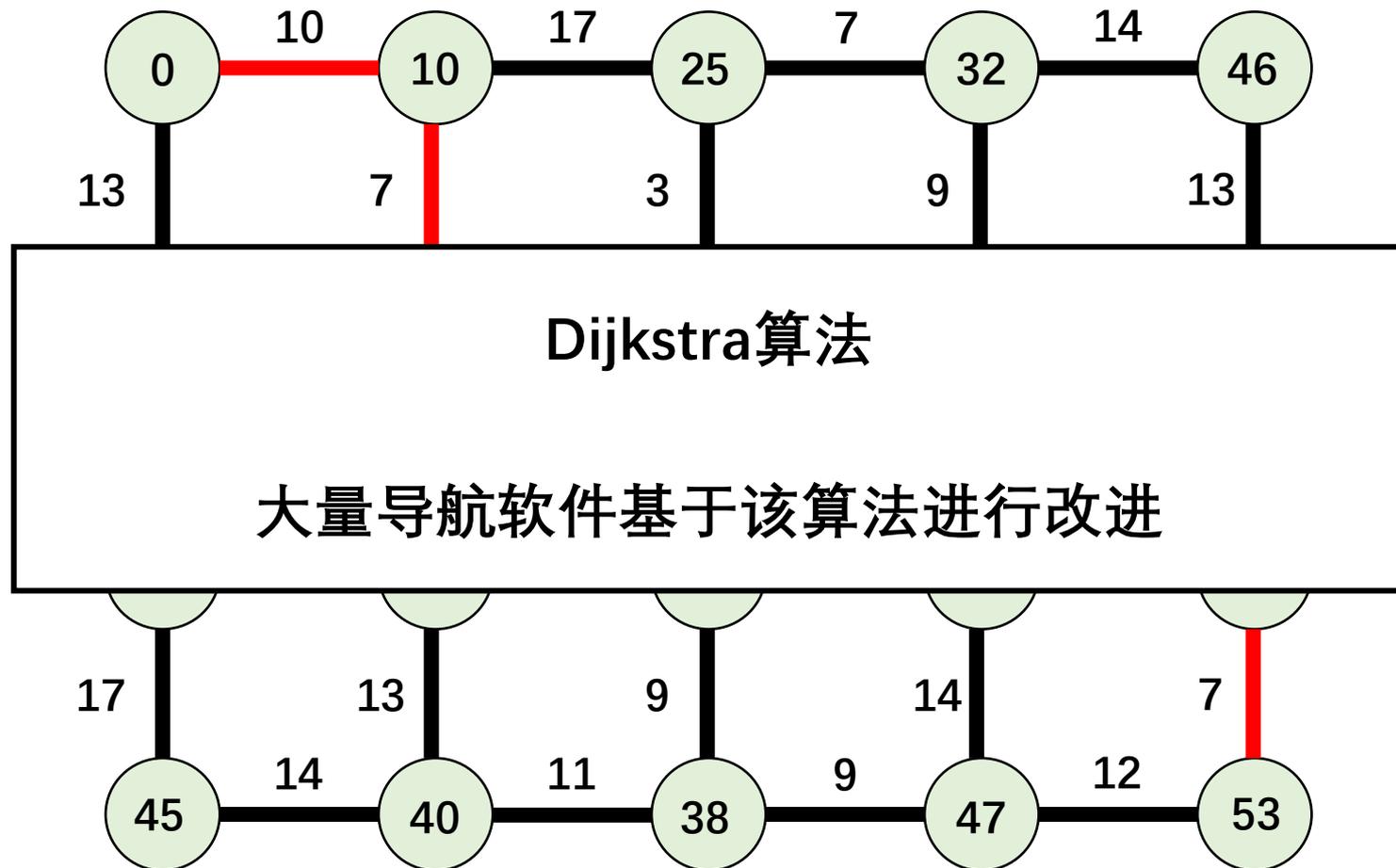
寻找最短路径



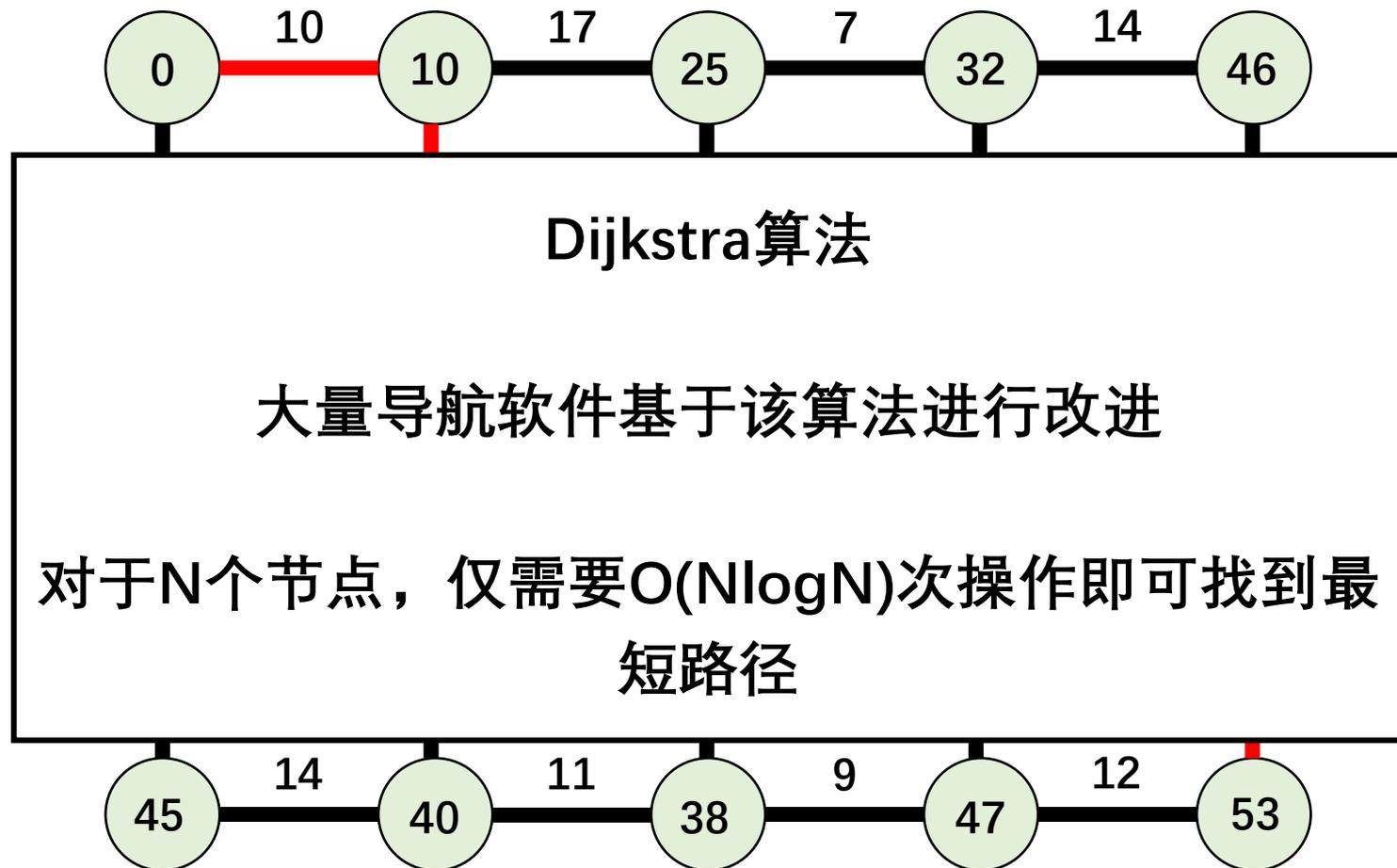
寻找最短路径



寻找最短路径



寻找最短路径



第三阶段：抽象与哲学

白日依山尽，黄河入海流

直白简洁的描述

欲穷千里目，更上一层楼

意味深长的哲理

```
int arr[] = {1, 2, 3, 4, 5};  
int sum = 0;  
for (int i = 0; i < 5; ++i) {  
    sum += arr[i];  
}
```

使用循环进行累加

使用STL容器和迭代器，类型无关，避免越界，避免修改

```
int arr[] = {1, 2, 3, 4, 5};  
int sum = accumulate(begin(arr), end(arr), 0);
```

课程主题

掌握一门计算机程序语言的基础语法

能写出语法正确的代码 - 能通过编译器检查

能写出尽量“漂亮”的代码 - 能通过你的审美标准

使用该语言解决简单到中等难度的问题

将问题转换为合适的算法问题并使用代码解决

体会计算机编程抽象

部分理解编程语言的设计原理

部分理解算法问题的转换思路

课程主题

掌握一门计算机程序语言的基础语法

能写出语法正确的代码 - 能通过编译器检查

能写出具有“漂亮”的代码 - 能通过作业的审美标准

使用该语言

将问题转化

体会计算机编程的优雅

搭建与计算机**沟通**的桥梁

构筑计算机**系统**观

打好研究计算机**科学**的基础

决

部分理解编程语言的设计原理

部分理解算法问题的转换思路

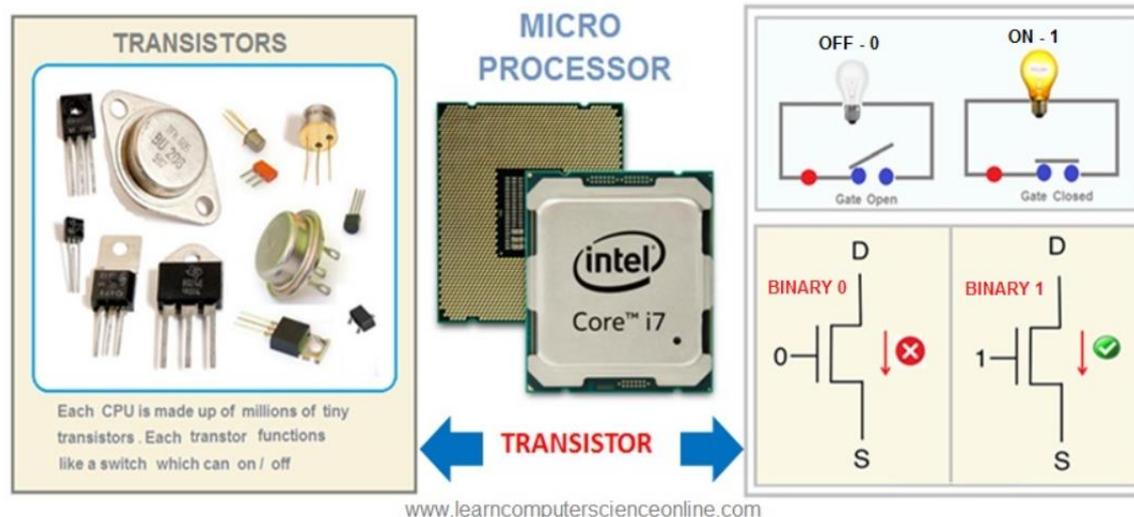
认识计算机编程语言

计算机编程语言层次 - 1

计算机由大量晶体管组成

二进制编码是天然的计算机编程语言

机器语言： 对电路的直接描述



计算机编程语言层次- 2

大量重复电路完成确定性功能

加法，乘法，读取内存

将确定性功能抽象为更高级的**指令**

汇编语言：描述CPU需要执行的指令

不同的CPU对确定性电路的抽象不同，指令集不同

x86汇编

```
MOV EAX, 5 ;
MOV EBX, 3 ;
ADD EAX, EBX ;
```

ARM汇编

```
LDR R0, =5 ;
LDR R1, =3 ;
ADD R0, R0, R1 ;
```

计算机编程语言层次- 2

大量重复电路完成确定性功能

加法，乘法，读取内存

将确定性功能抽象为更高级的指令

汇编语言

只面向CPU能力：**低级编程语言**

不同的CPU对确定性电路的抽象不同，指令集不同

x86汇编

```
MOV EAX, 5 ;
MOV EBX, 3 ;
ADD EAX, EBX ;
```

ARM汇编

```
LDR R0, =5 ;
LDR R1, =3 ;
ADD R0, R0, R1 ;
```

计算机编程语言层次- 3

实际用计算机解决问题的思路

根据问题寻求抽象方案 → 将方案映射到计算机的抽象上

有100个数字，我如何找到中位数？

解决方案抽象：对数字进行排序，找到排在中间的数字

方案映射：使用数组存储所有数字，排序后使用内存地址访问

```
int find_median(std::vector<int>& nums) {
    std::sort(nums.begin(), nums.end());
    return (nums[size / 2 - 1] + nums[size / 2]) / 2.0;
}
```

计算机编程语言层次- 3

实际用计算机解决问题的思路

根据问题寻求抽象方案 → 将方案映射到计算机的抽象上

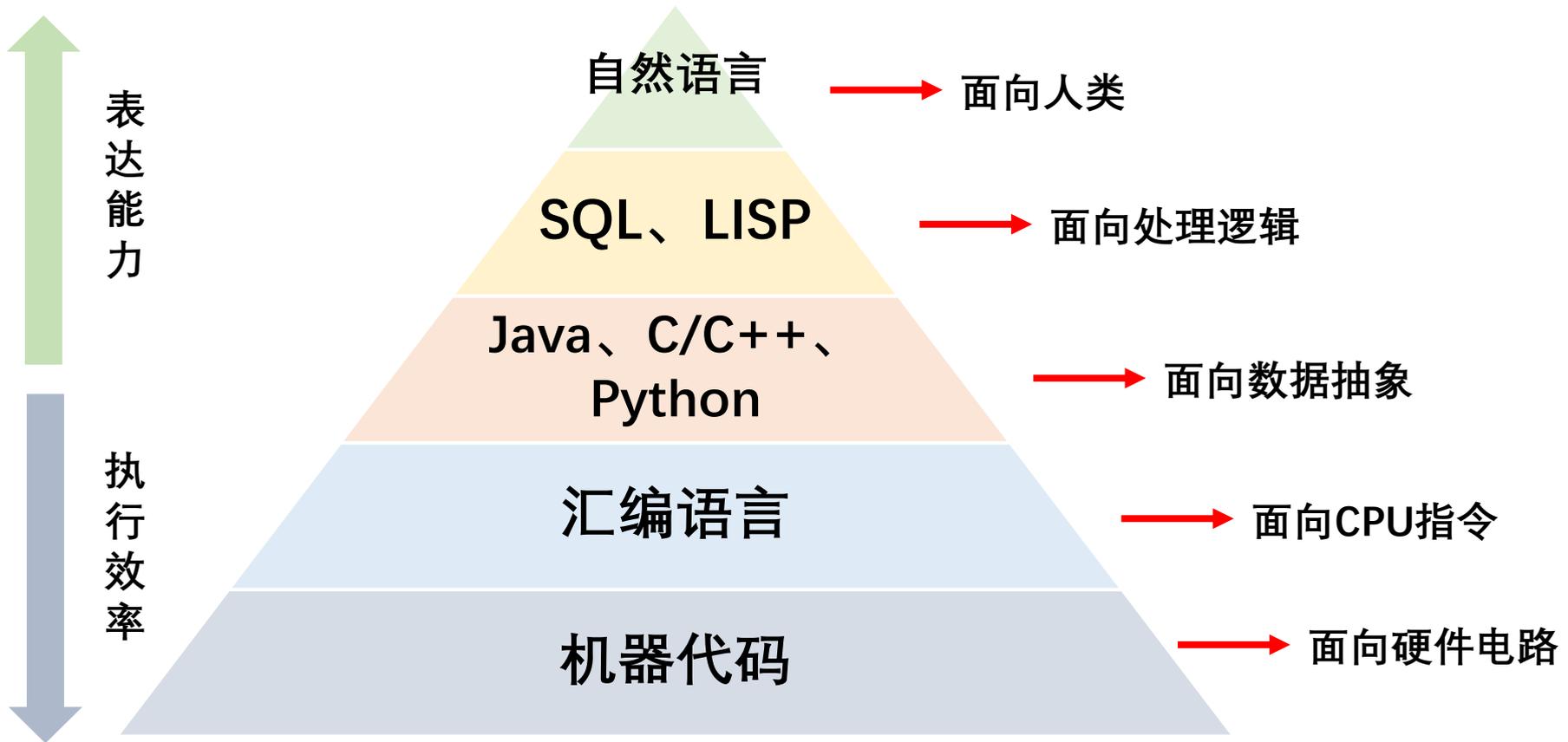
有100个数字 我如何找到中位数？

同时面向解决方案抽象和系统能力抽象：**高级编程语言**

方案映射：使用数组存储所有数字，排序后使用内存地址访问

```
int find_median(std::vector<int>& nums) {  
    std::sort(nums.begin(), nums.end());  
    return (nums[size / 2 - 1] + nums[size / 2]) / 2.0;  
}
```

计算机编程语言层次 - 4



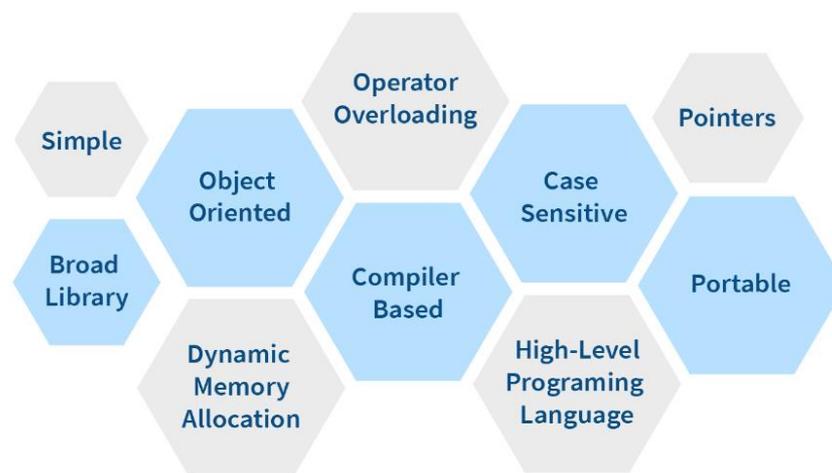
认识C++语言

由C语言发展而来，++是C语言的自增操作符

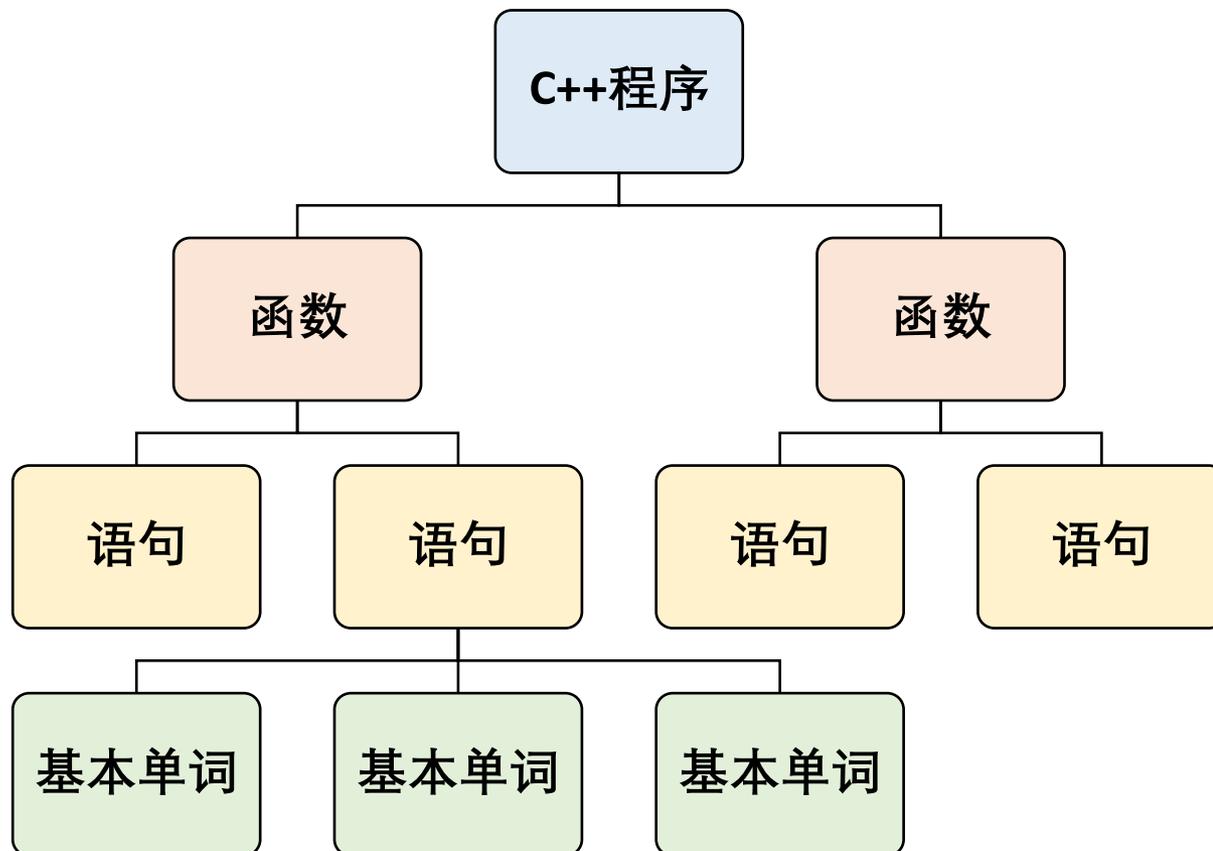
具有多种编程范式：面向过程、面向对象、泛型编程等等

执行效率高，适用于系统级编程：操作系统、高性能计算

抽象层次适中：便于学习理解计算机底层体系结构



C++程序基本结构



C++语句 (statement)

C++使用语句 (statement) 表示需要进行的动作

变量类型 (关键字) 变量名 语句由分号分割

声明一个变量

`int num;`



在内存中预留
一片存储空间

对变量进行赋值

操作符 常量

`num = 3 * 5;`



常量计算后，更新变量
对应的存储空间的内容

`int num = 3 * 5;`

C++函数 (function)

也被称为**子过程** (sub-procedure)

接收0到多个参数，返回0个或1个值

返回类型 函数名 参数列表

void表示返回类型为空 不接收参数

```
int square(int x)
{
    return x * x;
}
```

```
void print()
{
    cout << "Hello XJTU!";
}
```

大括号括起函数体

返回关键字

不需要返回

你的第一个C++程序

预编译指令，会被编译器自动替换



```
#include <iostream>

int main()
{
    using namespace std;
    cout << "Hello, world!";
    cout << endl;
    return 0;
}
```

主函数，程序
的唯一入口

向标准输出打印字符串 →

向标准输出打印换行 →

你的第二个C++程序

```
#include <iostream>

int main()
{
    using namespace std;

    int carrots; // declare a variable
    carrots = 25; // assign the variable
    cout << "I have ";
    cout << carrots; ← 输出变量
    cout << " carrots.";
    cout << endl;

    carrots = carrots - 1; // modify the variable
    cout << "Now I have " << carrots << " carrots." << endl;
    return 0;
}
```

使用//进行单行注释

输出变量

连续使用<<

你的第三个C++程序

使用计算公式打印0-300华
氏温度与摄氏温度的对照表

$$C = (5/9) * (F-32)$$

```
int main()
{
    int celsius;
    int fahr = 0;
    int upper = 300, step = 20; // initialize two vars

    while (fahr <= upper)
        celsius = 5 * (fahr-32) / 9;
        cout << "F: " << fahr;
        cout << ", Celsius: " << celsius << endl;
        fahr = fahr + step;
}
return 0;
}
```

循环语句 →

← 循环条件，只要满足就一直循环

← 循环体，用大括号括起，每次循环执行

我会写C++程序了！



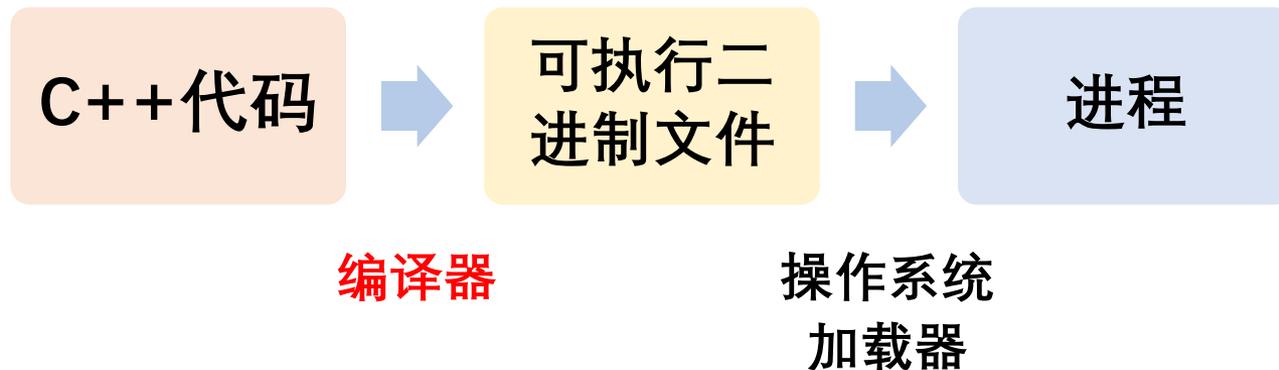
.....但是怎么运行？

编译执行程序 - 1

CPU只能执行汇编语言

需要先将高级语言转换为汇编语言：**编译 (compile)**

操作系统加载并执行汇编语言代码



*解释型语言依靠解释器进行执行

编译执行程序 - 2

在命令行中输入编译命令并执行

以Linux下的g++编译器为例

```
$ g++ hello.cpp
```

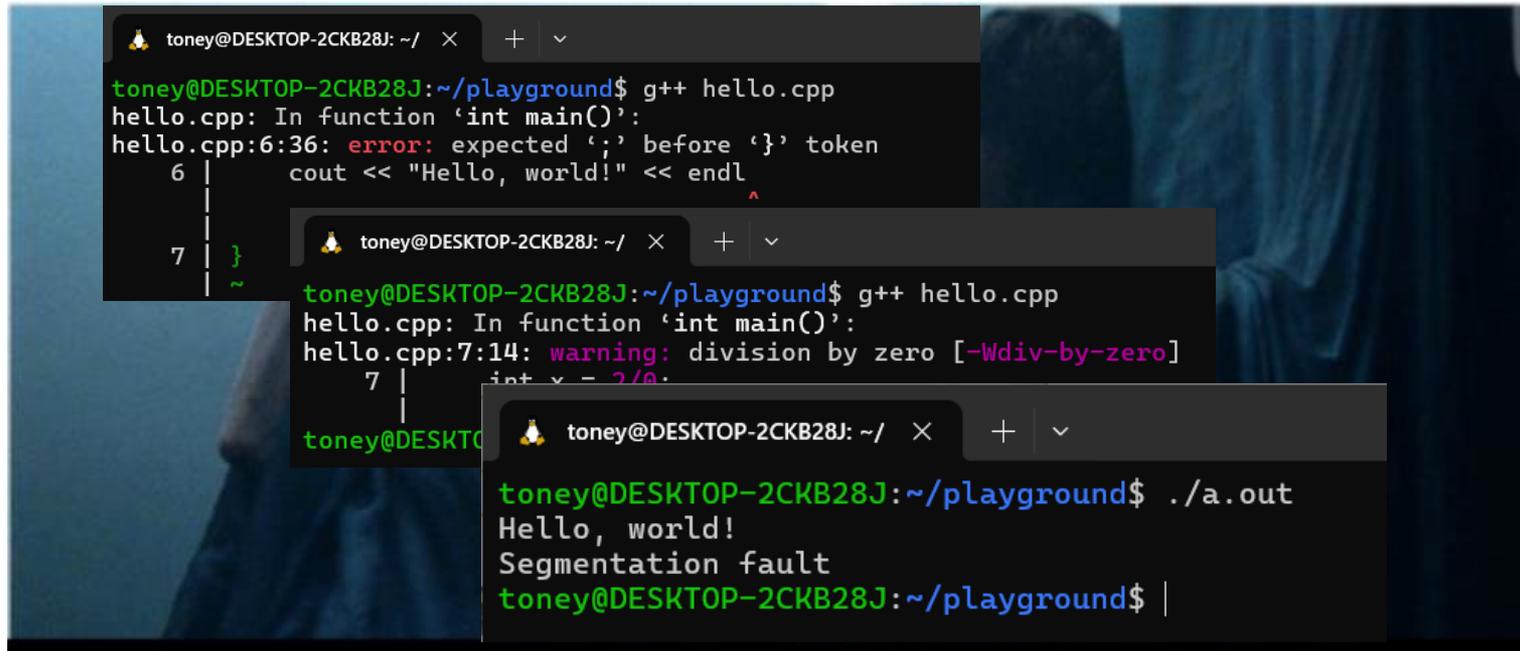
↑
编译器程序

↑
源代码文件

```
$ ./a.out
```

↑
生成的可执行文件默认在当前目录./
默认文件名为a.out

我会写C++程序了！



```
toney@DESKTOP-2CKB28J: ~/   
+   
v   
toney@DESKTOP-2CKB28J:~/playground$ g++ hello.cpp   
hello.cpp: In function 'int main()':   
hello.cpp:6:36: error: expected ';' before '}' token   
6 |         cout << "Hello, world!" << endl   
   |                                     ^   
7 |     }   
   |     ~   
toney@DESKTOP-2CKB28J:~/playground$ g++ hello.cpp   
hello.cpp: In function 'int main()':   
hello.cpp:7:14: warning: division by zero [-Wdiv-by-zero]   
7 |         int x = 2/0;   
   |                
toney@DESKTOP-2CKB28J:~/playground$ ./a.out   
Hello, world!   
Segmentation fault   
toney@DESKTOP-2CKB28J:~/playground$ |
```

程序错误

编译错误：语法错误或可以被静态推断出的语义错误

```
cout << "Hello, world!"
```

```
int x = 2 / 0;
```

运行时错误：运行时导致程序崩溃的错误

从标准输入接收字符



```
int x, y;  
cin >> y;  
x = 2 / y;
```

调试程序

编译错误：检查编译器给出的错误提示

```
toney@DESKTOP-2CKB28J: ~/playground$ g++ hello.cpp
hello.cpp: In function 'int main()':
hello.cpp:6:36: error: expected ';' before '}' token
   6 |         cout << "Hello, world!" << endl
     |                                     ^
     |                                     ;
   7 |     }
     |     ~
```

运行时错误：使用调试工具跟踪程序运行状态

```
Reading symbols from ./a.out...
(gdb) r
Starting program: /home/toney/playground/a.out
0

Program received signal SIGFPE, Arithmetic exception.
0x000055555555220 in main () at hello.cpp:8
8         x = 2 / y;
```

工欲善其事，必先利其器

编程工具链 (toolchain)

编辑器：“写”代码的工具

编译器：将代码转换为可执行文件

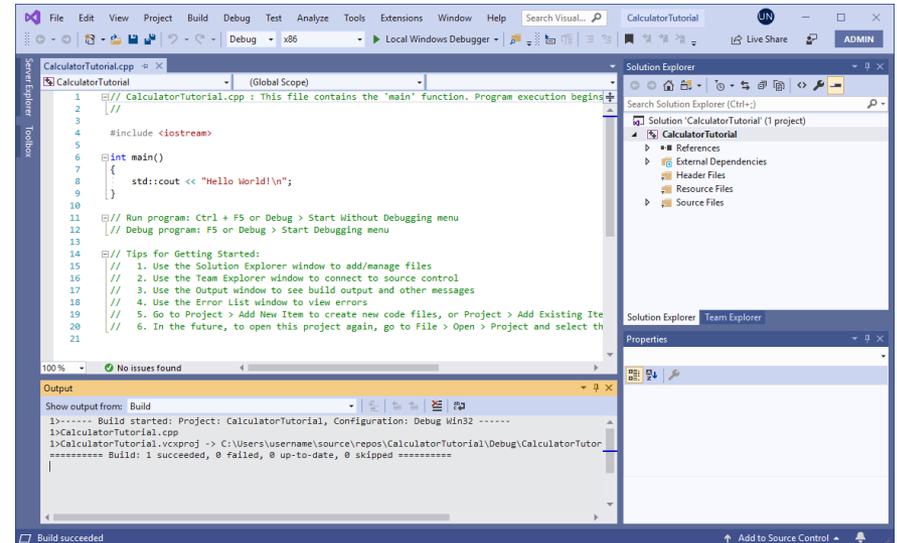
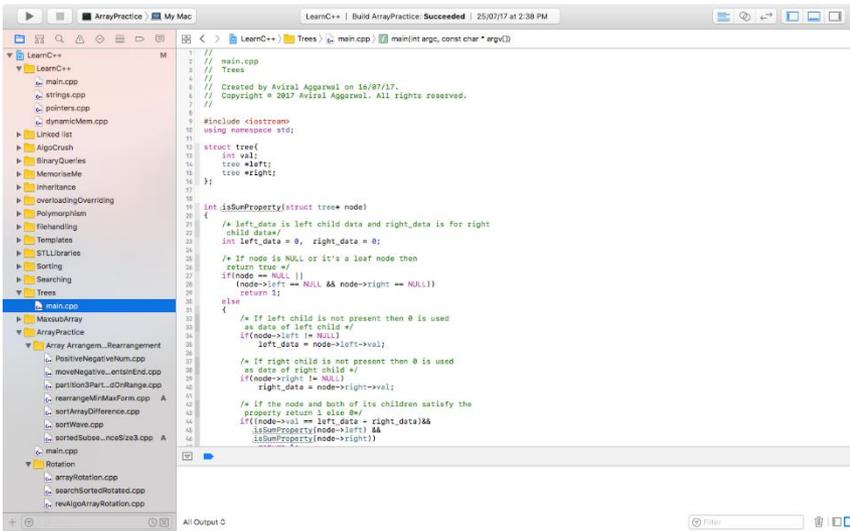
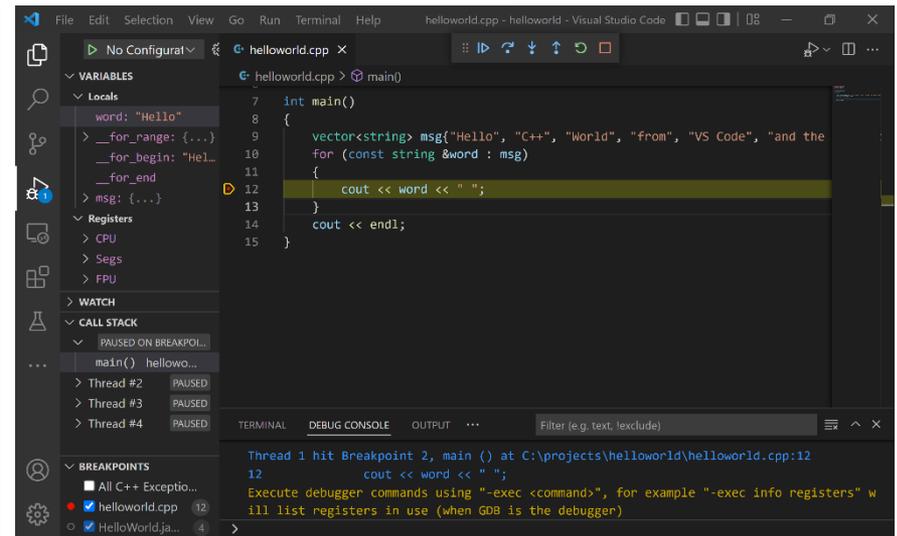
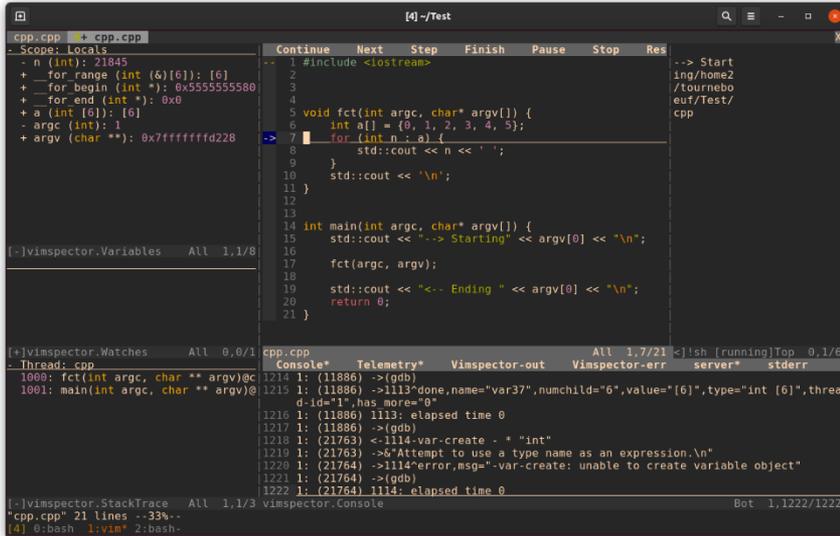
调试器：跟踪、暂停、查看代码内部状态

集成开发环境 (IDE)

把上面三者集成在同一个软件框架中

简便，高效，可视化

本课程推荐手动使用toolchain，建立底层系统观



工欲善其事，必先利其器

差生文具多

Welcome and

