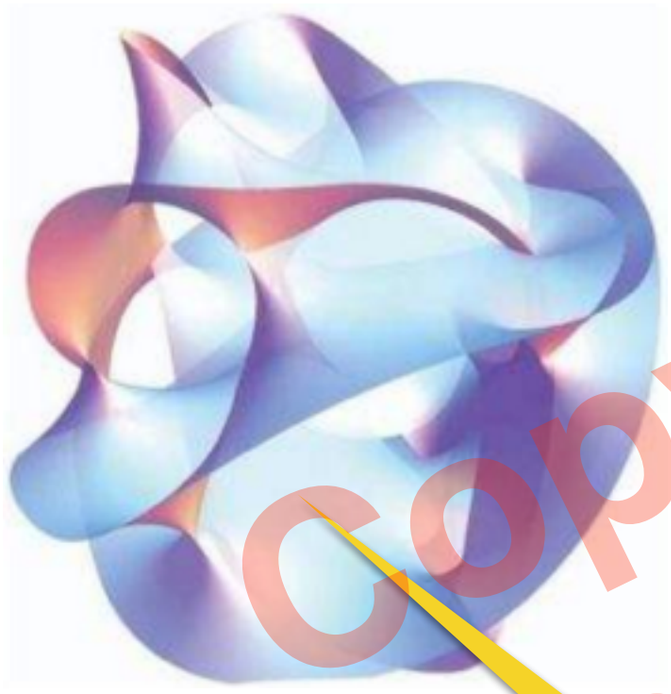
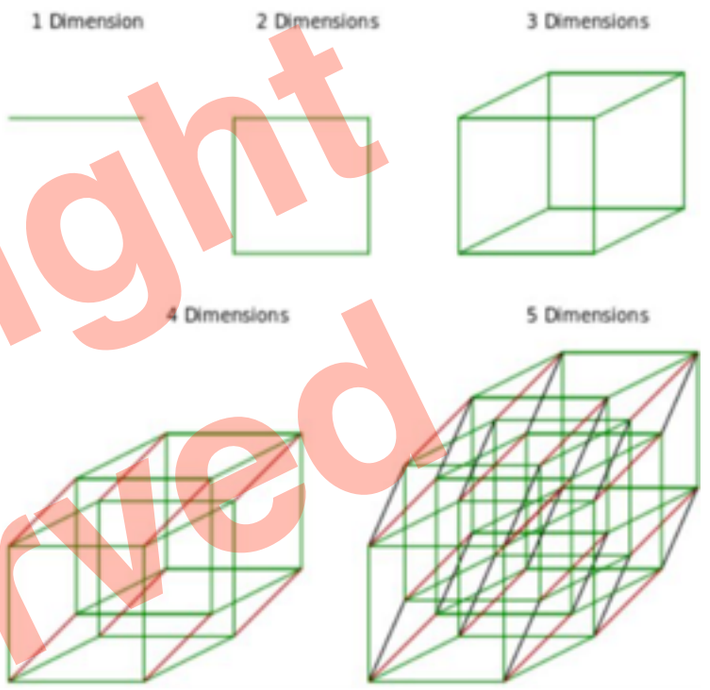
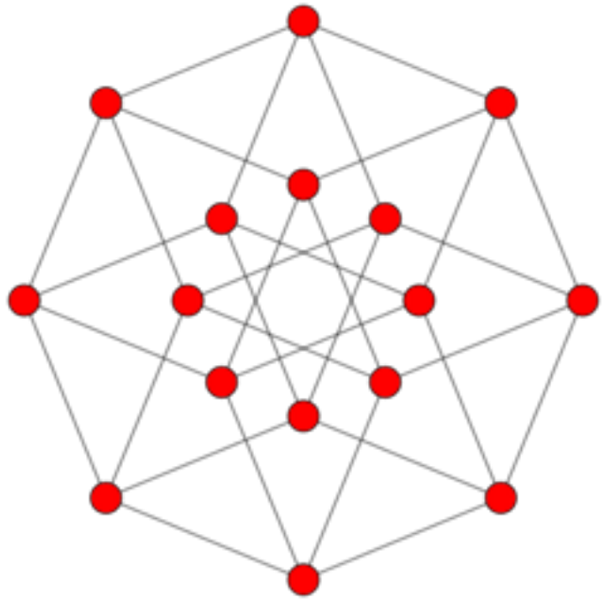
作业回顾：非编程题

1.高维空间



卡拉比-丘成桐空间



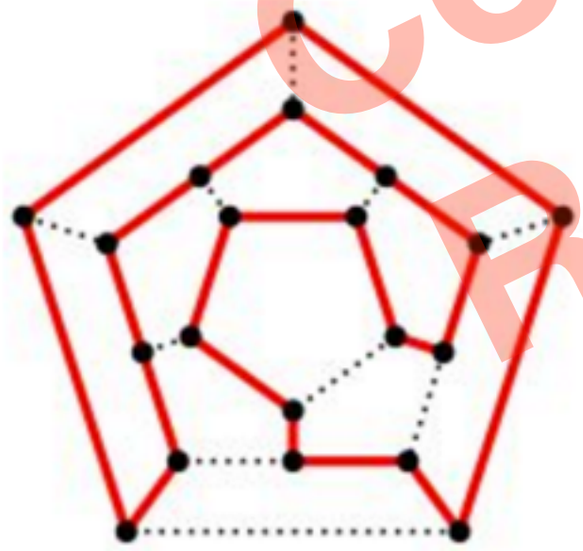


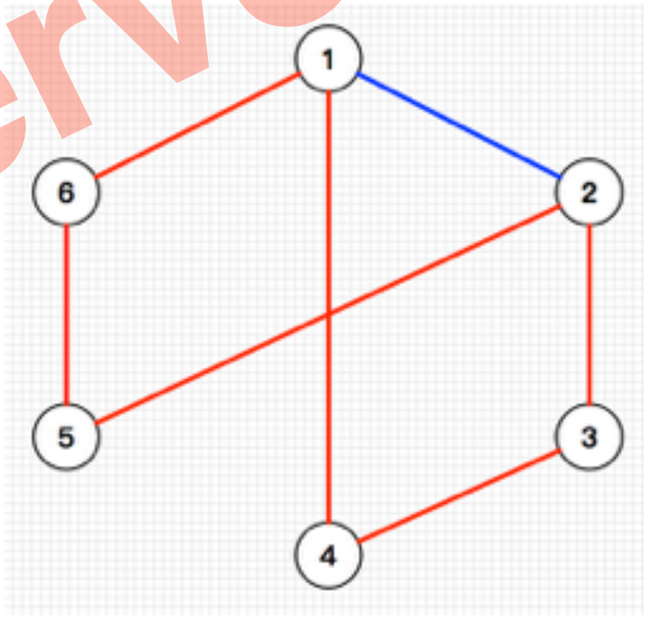
作业回顾：非编程题

3.哈密尔顿回路

Hamilton回路是**经过一个图的每个顶点一次且仅一次，并且回到起始点**的路径
（下期课上你们会学到图论算法）

每个数字看做一个点，相加等于质数的两个点之间连线
那么素数环就是这个图上的一个哈密尔顿回路





作业回顾：编程题

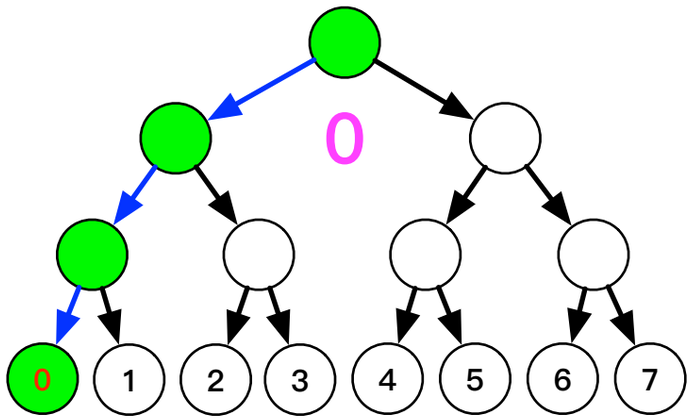
2.fallingballbetter.cpp

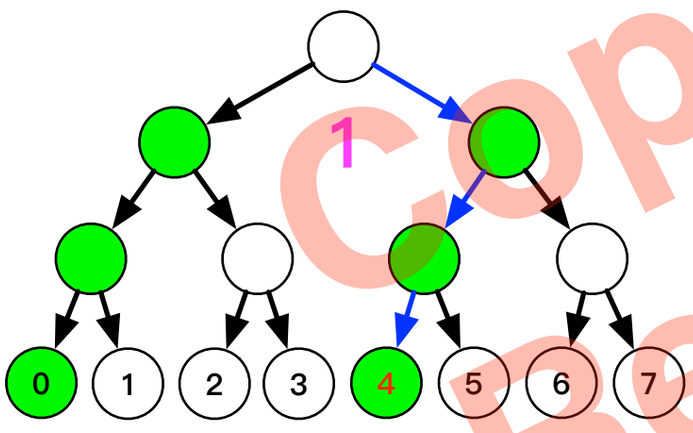
4.primeringbetter.cpp

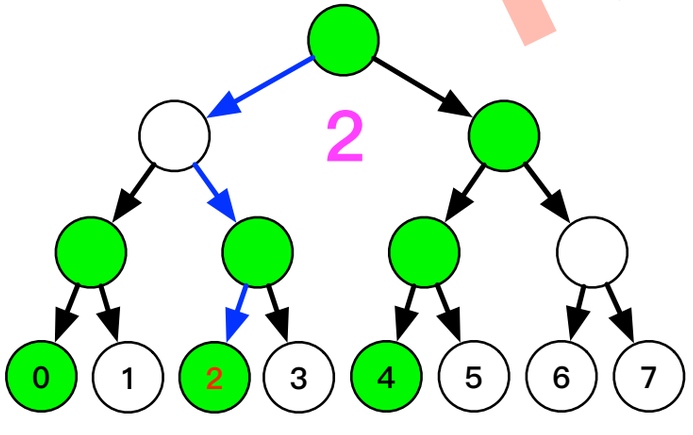
5.eightqueen.cpp

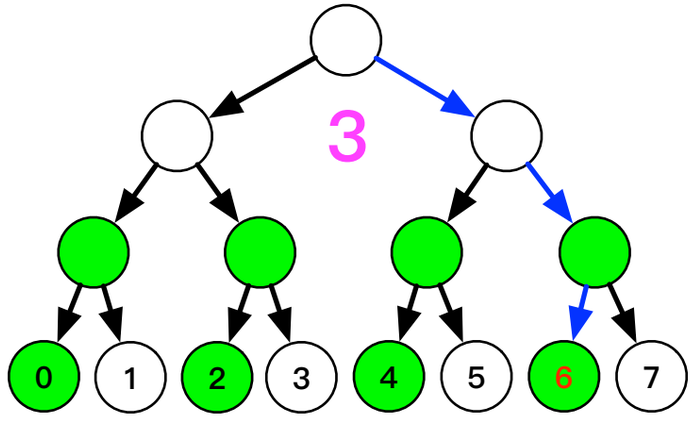
Welcome to Xcode

作业回顾：小球下落问题-思路









→ 先把灯重新编号，最后一排从0开始（位移2^D号）

→ 小球从0开始编号（位移1号）

→ 偶数号球落向左子树，奇数号球落向右子树，形成自相似

→ 状态：f(D,n)

→ 状态转移：

向左：f(D,2k)=f(D-1,k)=f(D-1,2k>>1)

向右：$f(D,2k+1)=f(D-1,k)+2^{D-2}=f(D-1,(2k+1)>>1)+1<<(D-2)$

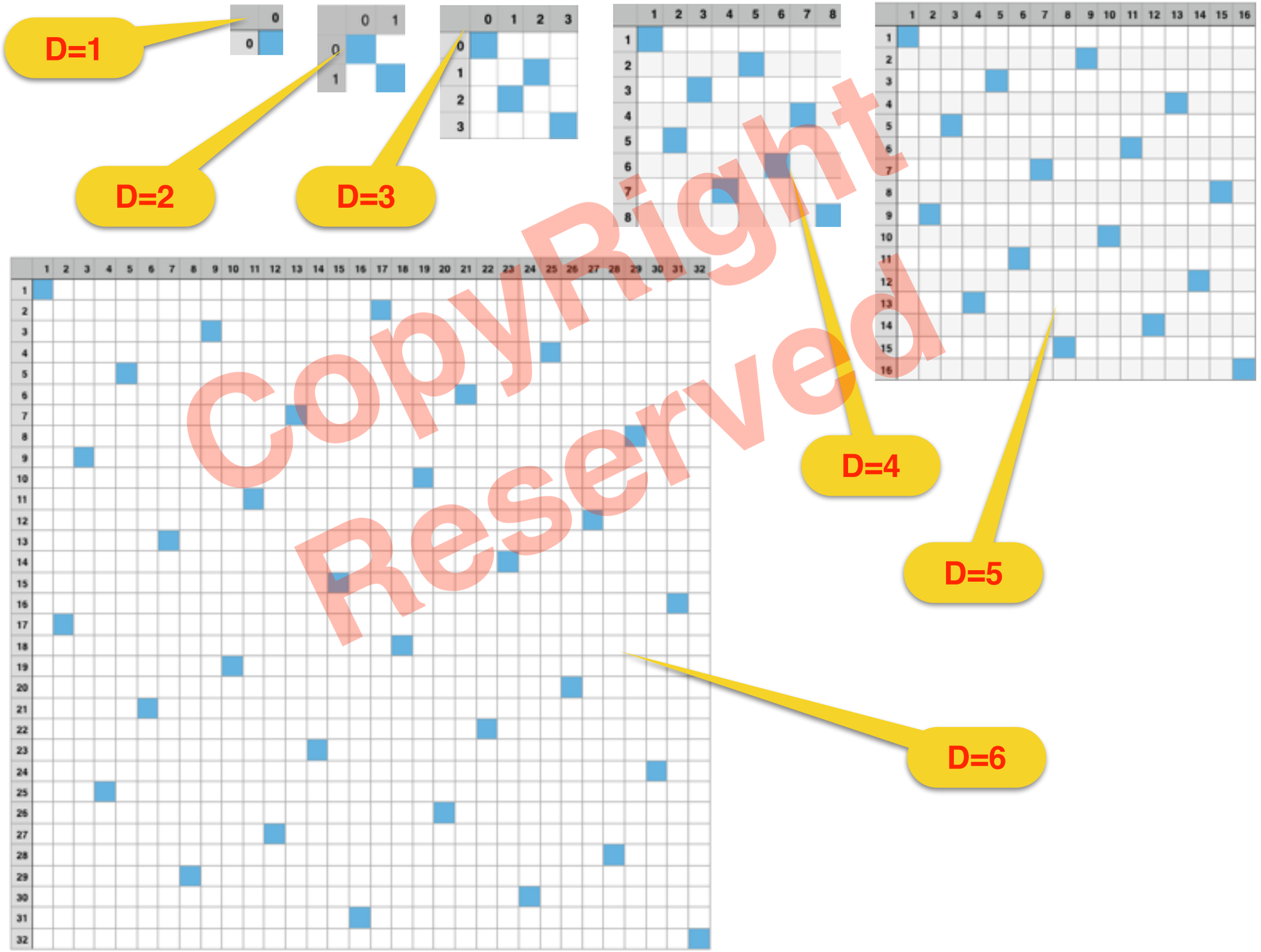
→ 边界：f(D,0)=0

→ 从二进制角度看，就是把D-1位二进制数n逆序

f(4,2)=f(4,010)=010=2

f(4,3)=f(4,011)=110=6

作业回顾：小球下落问题-解的规律



CS100 算法入门

高级话题

文件读写

```
15  void fileIODemo(){
16      // 打开输入文件，注意改成你自己的文件的绝对路径
17      // 如果使用相对路径，Xcode会有特殊的相对路径根目录
18      freopen("/Users/gexiao/Documents/fudan-course/week8/syntax/input.txt", "r", stdin);
19      int a[10];
20
21      for(int i=0;i<10;i++){  // 输入
22          cin>>a[i];
23          a[i]=addOne(a[i]);  // 调用util.hpp里的函数，实现在util.cpp
24      }
25
26      freopen("/Users/gexiao/Documents/fudan-course/week8/syntax/output.txt", "w", stdout);  // 打开输出
            文件
27      for(int i=0;i<10;i++){  // 输出
28          cout<<a[i]<<" ";
29      }
30      cout<<endl;
31  }
```

→ stdin/stdout称为标准输入输出，也就是键盘和屏幕
其实还有一个stderr称为标准错误，不过你们用不到

→ 为什么输入输出的语句并没有变化？
这种技术成为重定向，把输入输出抽象成对一个虚拟端的读写，可以是控制台，也可以是文件。这是工程中一个很重要的概念。有兴趣可以自己去了解一下

→ 注意！竞赛时一定要看清是使用标准输入输出还是文件输入输出

→ 有些题目的输入比较多，每次手输一遍很麻烦
→ 这时可以考虑把数据存在文件里，让程序从文件输入数据，节省时间

跨文件调用

cs002 〉 cs002 〉 basic.hpp 〉 No Selection

```
11
12  #include <stdio.h>
13
14  void fileIODemo();
```

```
21  #include "basic.hpp"     // 引用头文件
22
23  using namespace std;
24
25  int main(){
26      fileIODemo();    // 调用头文件里的函数
27  }
```

→ 一定要把所有代码放在同一个文件吗？当然不是

→ 你可以创建一个.hpp/.h文件（称为头文件），存放方法签名，然后其他代码通过#include导入头文件，就可以跨文件调用了

→ 之前你们使用的stdio.h, iostream, vector之类的，其实就是C++系统提供的头文件

→ 注意！竞赛时只要写一个代码文件，不要分文件

→ 为什么不把代码都写在一起?

1.实际的IT项目代码动辄几万行，如果不分门别类代码根本无法维护

2.大型项目通常是团队开发，每个人负责各自的代码文件

3.不同功能性质的代码组织在不同的文件，方便代码重复使用，也方便测试。这种思路称为模块化（IT从业人员必备思想）

STL-排序

→ STL是C++标准模板库（Standard Template Library），提供了很多现成代码
→ 其实你们学的算法很多都是现成的，比如排序
→ 这是使用C++的一个优势，避免每次自己写标准算法，节省大量竞赛时间

```cpp
#include <algorithm>

int main(){

    int a[10]={2,4,1,23,5,76,0,43,24,65};
    sort(a,a+10);    // 调用C++标准库的排序函数
    for(int i=0;i<10;i++){
        cout<<a[i]<<" ";
    }
    cout<<endl;
}
```

```
0 1 2 4 5 23 24 43 65 76
Program ended with exit code: 0
```

STL-优先队列

```cpp
#include <iostream>
#include <vector>
#include <queue>

using namespace std;

int main(int argc, const char * argv[]) {
    /* 14
     53 542 5 63 14 217 154 748 616 742 111 412 452 134*/

    int n;
    cin>>n;

    priority_queue<int> heap=priority_queue<int>();
    for(int i=0,x;i<n;i++){
        cin>>x;
        heap.push(x);      // 依次入堆
    }

    for(int i=0;i<n;i++){
        cout<<heap.top()<<" ";   // 依次出堆
        heap.pop();
    }
    cout<<endl;
    return 0;
}
```

老师你忽悠我?

两句话就解决的你讲一节课?

→ priority_queue是stl提供的标准优先队列，实现堆排序就秒杀了

→ 想查找树等其他标准算法，网上很容易找到现成代码，加个头文件就能用

学算法的意义

→ 算法告诉你解决问题的思路/模式，帮助你自己解决遇到的新问题

真爱难不倒技术宅:美数学家利用大数据找到女朋友_爱活网 Evolife.cn

2014年1月24日 - 可以说没有什么问题能够真正难倒他们——即使是用数学方法找到一名女朋友,...者,于是他决定利用自己收集的大数据进行更加精确的配对,从而让自己找到真爱...

www.evolife.cn/html/20... - 百度快照 - 96%好评

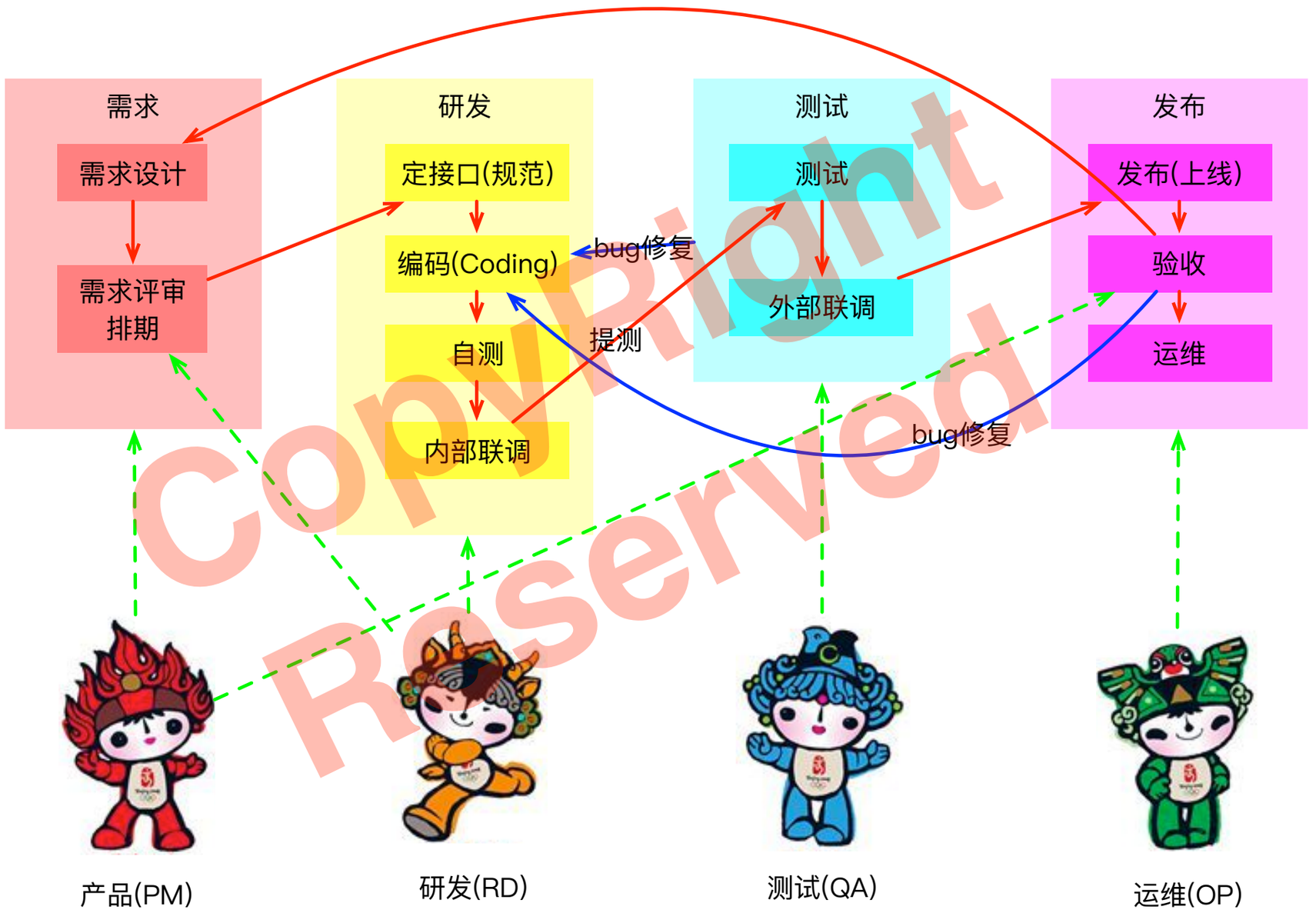
→ 算法里蕴含了哲学，传达对这个世界的一种认知，帮助你建立科学的世界观（有点矫情，但是事实）

→ 基本算法的组合会产生神奇的效果。同时高级别的竞赛（NOI/IOI）经常需要组合超过一种算法

→ 你们不学算法我没法赚钱💰

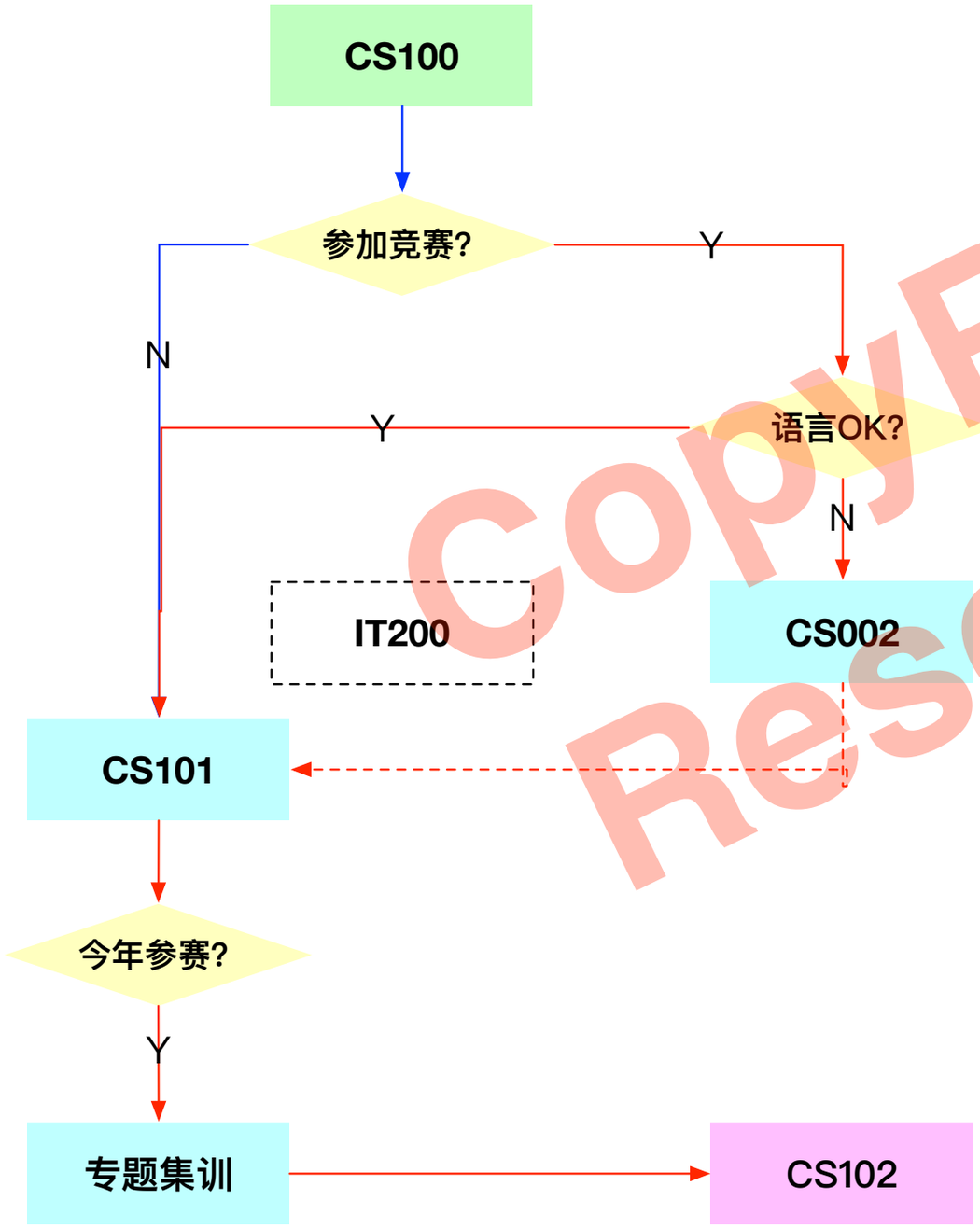
那为什么还要学算法?

开发大型项目



大互联网行业欢迎你入行(keng)😝

重播广告：暑假后续课程



→ **CS002（16学时）**

C++语言（零基础）

→ **CS101（32学时）**

更多竞赛算法，基本全部覆盖竞赛范围

（动态规划，图论，模式匹配，散列……）

→ **专题训练（12学时）**

竞赛流程，应试技巧，初赛(笔试)应对，真题实战

→ **CS102（32学时）**

更多真题实战，模拟考试，etc

→ **IT200（16学时）**

Scratch入门

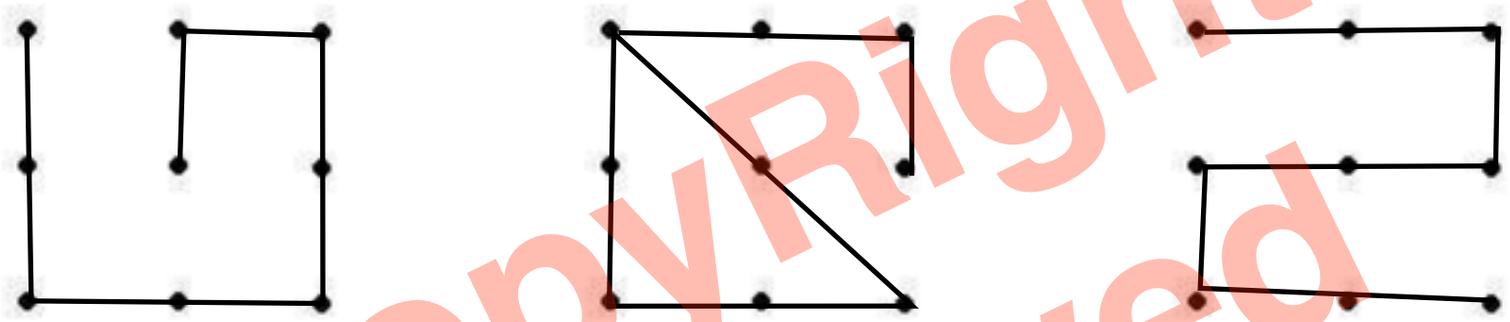
→ **什么叫语言OK?**

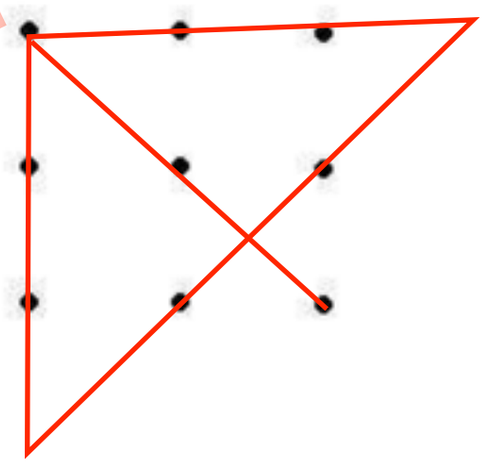
可以不借助外力独立编码

来点好玩的：思维定势

四线问题：

使用4条连续的线段连接3x3共9个点





来点好玩的：芝诺佯谬（有限与无限）

→ **这是古希腊一个著名的思辨课题**

→ **芝诺是古希腊哲学家，他试图证明阿喀琉斯（古希腊神话中善跑的英雄）追不上一只乌龟**

→ **假设阿喀琉斯落后乌龟10米**

阿喀琉斯跑10米，乌龟跑1米

阿喀琉斯又跑1米，乌龟又跑0.1米

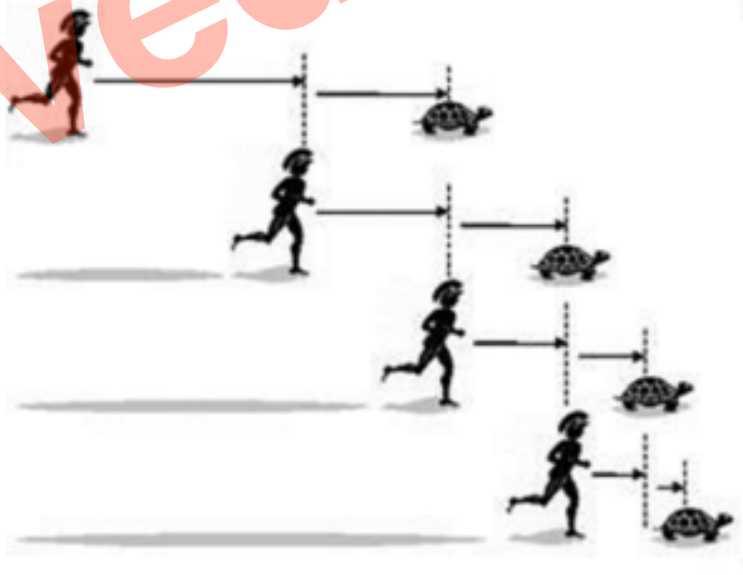
阿喀琉斯又跑0.1米，乌龟又跑0.01米

……（永远追不上）

→ **这个问题的解答是如下的无穷级数公式**

10+1+0.1+0.01+……=11.1111……=11+1/9

→ **无限多个数加在一起，可以得到一个有限大的数**



来点好玩的：理发师悖论（集合论）

→ 集合你们多少了解，就是一些事物的组合
→ 有些人听说过理发师悖论，下面是一种更有(dan)趣(teng)的版本

→ 对一个自然数，可以通过一些汉字和英文字母来描述
例如1000可以描述为“一千”、“十的三次方”、“最小的四位数”等等
→ 现在构造一个集合A，表示所有可以用不超过20个汉字/字母描述的自然数

→ 因为汉字/字母的总数是有限的（比如有x个），那么A的元素也是有限个（不超过x^{20}）
→ 那么A里面一定有一个最大的数，比如叫做a
→ 那么b=a+1不在A里面

→ 但是可以这么描述b：不在集合A中的最小的自然数（13个）

→ 为了解决这个问题，两个蛋疼的数学家建立了一整套公理体系来规定怎样定义集合才是合理的，称为策梅洛-弗兰克尔（Zermelo-Fraenkel）体系，你们以后如果学数学分析可能会学到

有没有感觉不相信人生了？😝

为什么要讲悖论

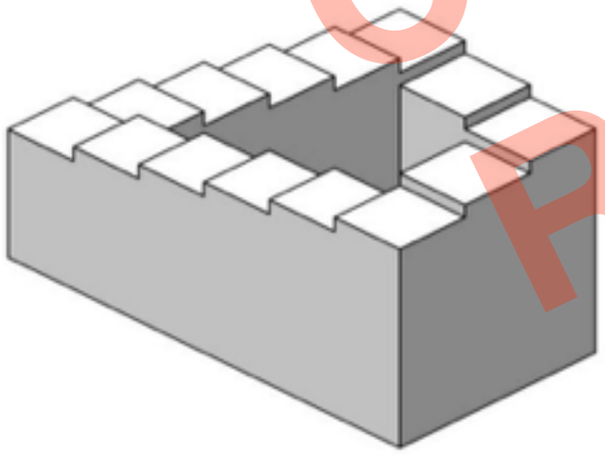
→ **很好玩**

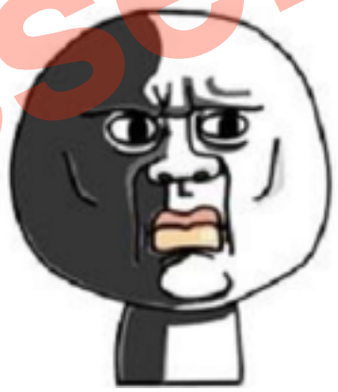
→ **生活中并不是所有事情都是非是即非**

真真假假的事情很多，有时候真的不必太认真

→ **如果你一定要认真，也许你就能成“家”了**

（数学家，哲学家...）





如果匹诺曹说：“我在说假话”，
那他的鼻子会不会边长？

最后讲一点哲学：宇宙观

→ 计算机科学和互联网行业某种意义上都是在创造一个**虚拟世界**(以实体世界为蓝本)

→ 当虚拟世界越来越接近真实(现在还差得远)
也许它真的就是一个**新的宇宙**

→ 所以编程某种意义上是一种**创世**
创造规则，创造世界

电商	商业
网游	娱乐
社交网站	交际
VR	生活
AI	人/智慧
互联网	社会
……	……

最后讲一点哲学：宇宙观

→ 反过来想，实体世界真的那么真实吗？

→ 也许宇宙就是某些神开发的一个很复杂的程序

→ 宇宙的语法是什么？

物理规律，数学规律

→ 所以学编程也能帮助我们更好地理解我们自己的宇宙

作业

当你选择一个方向并越走越远
你身边的人陆续放弃或离开
最后只剩你一个人时
你就是这个方向上最牛的人

你们之中将来一定会有人成就大事业
红包不是白拿的

坚持到最后的人都有红包

苟富贵，勿相忘！

结束语

我炒过马云的光荣事迹

曾经去蚂蚁金服面试，HR对我说：

在我们这里你是没有生活的（只有工作）

我说，谢谢那我还是不来了，因为

工作不是生活的全部

蚂蚁金服
ANT FINANCIAL

你们都提交了想去的地方

（纳尼，不记得了？第三周作业）

Just do IT！因为

学习也不是生活的全部

快乐比优秀更重要

扩展阅读：函数指针

```
31  void println(int& i){   // 输出函数
32      cout<<i<<endl;
33  }
34
35  void addone(int& i) { // 递增函数
36      i++;
37  }
38
39  void foreach(int data[],int n,void (*f)(int&)){ // 遍历函数，带一个函数变量类型参数
40      for (int i=0;i<n;i++) {
41          f(data[i]);
42      }
43  }
44
45  int main(){
46
47      int data[3]{1,2,3},i=5;
48      //      单个函数变量
49      void (*g)(int& )=addone;
50      g(i);
51      std::cout<<i<<std::endl<<std::endl;
52      //      函数变量传参
53      foreach(data, 3, addone);
54      foreach(data, 3, println);
55  }
```

```
6

2
3
4
Program ended with exit code: 0
```

→ 函数变量是一类特殊的变量，其值是一个函数（实际是指向函数代码在内存中的地址，所以也称为函数指针）。初步可以理解为函数的别名

→ 函数指针的作用与模板类似，用于统一一类抽象操作，例如遍历操作

→ 函数式编程已经日渐成为程序界的热门趋势，其思想就是函数指针，也就是把一种操作当做变量一样来看待