

# NOIP扫盲与备战

清华大学 计算机科学与技术系 王聿中

# 目录

扫盲

考察算法盘点

备考建议

考试技巧介绍

# Part1 扫盲

在这一部分，我们主要说NOIP.....

是什么？

- CCF NOI系列赛事&其他赛事简介
- NOIP的意义

考什么？

- 编程？
- 算法？ 数据结构？
- 其他？

# 一个OIER在一年里会经历什么？

## 主线

NOIP  
(11月)

省选  
(4月)

NOI  
(7月)

## 支线

CCF NOI  
系列赛事

清华集训  
(12月)

冬令营  
(1~2月)

CTSC  
(5月)

APIO  
(5月)

高校招生  
系列赛事

THUWC

PKUWC

1~2月

THUSC

PKUSC

5~6月

# 大家最关心的——哪些考试与升学直接相关？

有机会直接获得各高校邀约的比赛

- 高校招生系列赛事
- NOI

有机会获得**北京大学**邀约的成绩

- 在NOIP中取得高分（待考证）
- 在强省（浙江、湖南）省选中表现优异

**保送资格**

- 在NOI现场获得前50名的选手将入选国家集训队

# 暂时不想 / 没能力走到这一步？ NOIP之于你的意义？

## 止步于此

NOIP属于五大学科奥赛，是许多重点高校自主招生的入场券。

不错的业余爱好

培养思维

## 走得更远

打下扎实的基础并对OI产生正确的认知

培养高效的学习方法、习惯

验证自己的天赋

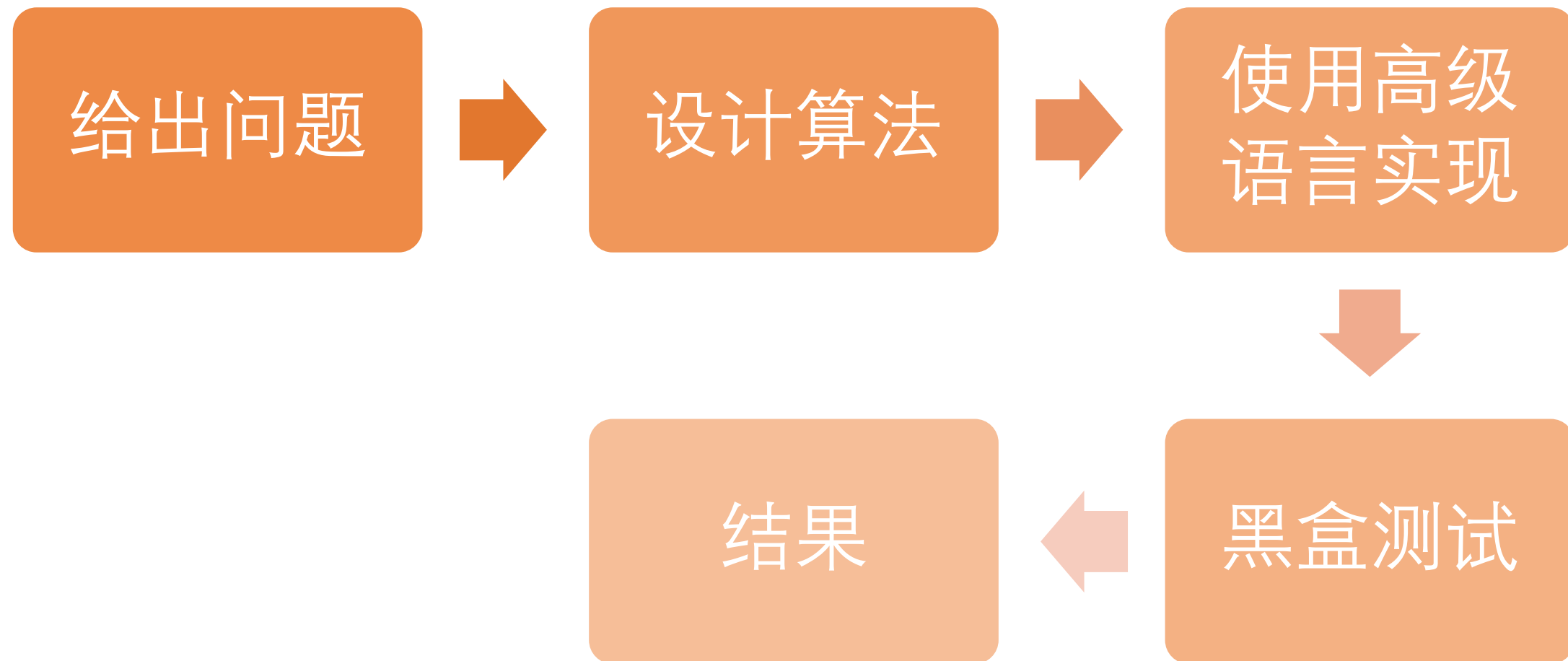
# NOIP考什么？

- 在回答这个问题之前，我们先看看OI比赛是什么





# 算法竞赛



# OI赛制

## 【子任务】

对于 10% 的数据，保证  $n \leq 9$ 。

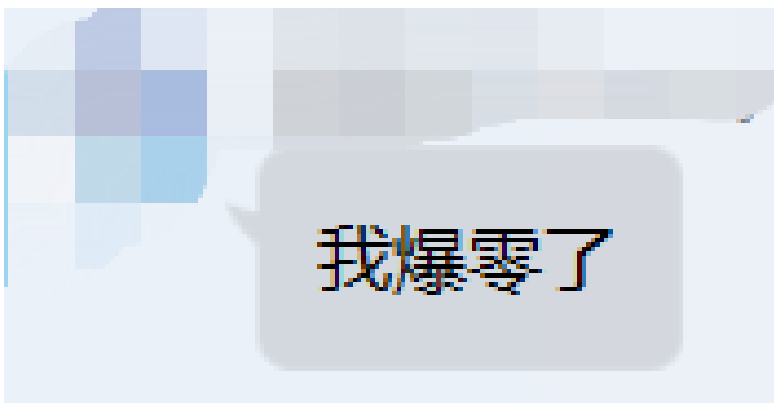
对于 30% 的数据，保证  $n \leq 18$ 。

对于 60% 的数据，保证  $n \leq 200$ 。

对于另外 20% 的数据，保证所有的  $w_i = 1$ ，保证所有的  $c_i = 1$ 。

对于 95% 的数据，保证  $n \leq 3,000$ 。

对于 100% 的数据，保证  $n \leq 5,000$ ， $w_i \leq 10^7$ ， $c_i < n$ 。



部分分

赛后评测

说了这么多，重点在哪？

设计算法

使用高级语言实现

互动1:状态道途是勳倦况的请举手

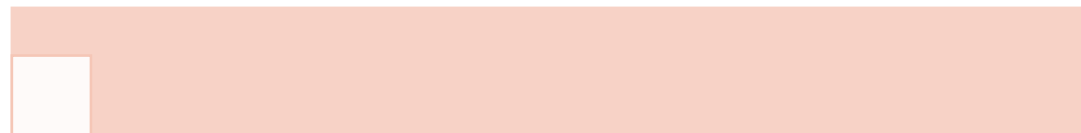
# 为什么要熟练掌握算法？

## 掌握



☐ 站在巨人的肩膀上

## 熟练

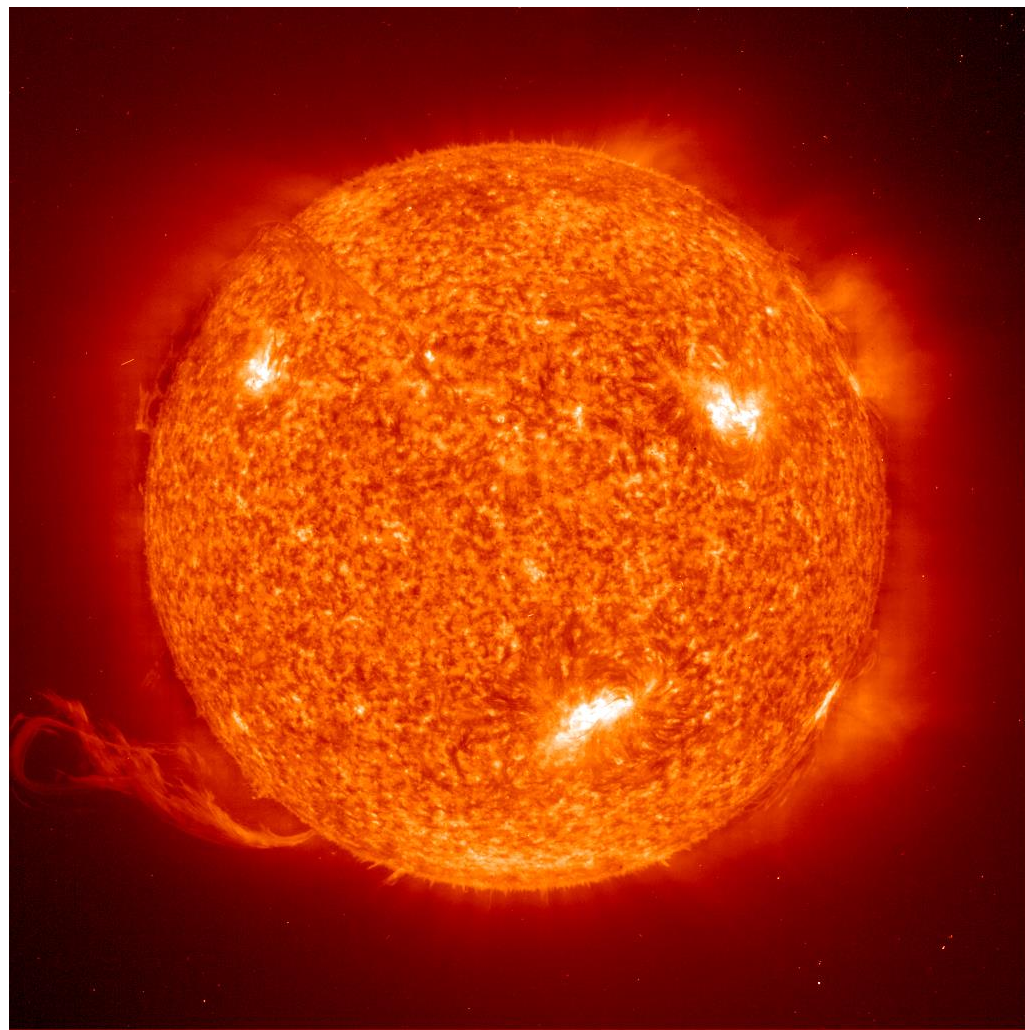


☐ 保证速度，节省时间

☐ 保证正确率，降低风险

互动2：知道递推数组么的请举手  
NOIP没考过450分的同学，请把手放下

已知 $1+1=2$   
求太阳的质量



掌握算法是远远不够的！

OI比赛侧重于考察  
选手的**思维水平**



嘴巴选手!

熟练掌握了算法并具备优秀的思维  
是否就能打遍天下了?

# 两个真实的案例

## 案例一（小Y同学在CTSC2016 Day1）

- 【0小时】 小Y同学读完P1并发现该题非常简单，遂开始写
- 【1小时】 小Y同学没有写完
- 【2小时】 小Y同学没有写完
- 【3小时】 小Y同学总算写完了，但样例都过不了
- 【4小时】 小Y同学没有调完
- 【5小时】 小Y同学没有调完

## 案例二（小X同学在NOI2015 Day2）

- 【0小时】 比赛开始，小X同学读完了题，发现都非常简单
- 【1小时】 小X同学做完了P1
- 【2.5小时】 小X同学做完了P2
- 【4.5小时】 小X同学做完了P3
- 【查分时】 100分

# 代码能力的体现



我上面的素质都具备  
平时刷题又快又准  
是不是万事俱备了?

# 案例三（闫书弈同学在CTT2017 Day1）

比赛开始!

这个P1好像能做，我来做做看吧!

4小时后

终于做完了

比赛结束后

这个P2和P3都好简单呀!

是呀是呀! 就是P1不会做!

OI成绩 = 做题水平 × 考试策略

(具体原话记不清了，大致是这个意思)

——陈立杰

## 案例四（小D同学在NOI2016 Day2）

你输出的实数在小数点后不能超过  $2p$  位，建议保留至少  $p$  位。

你的输出被判定为正确当且仅当你的输出与参考答案的绝对误差小于  $10^{-p}$ 。



那我就保留  $p+10$  位小数吧！

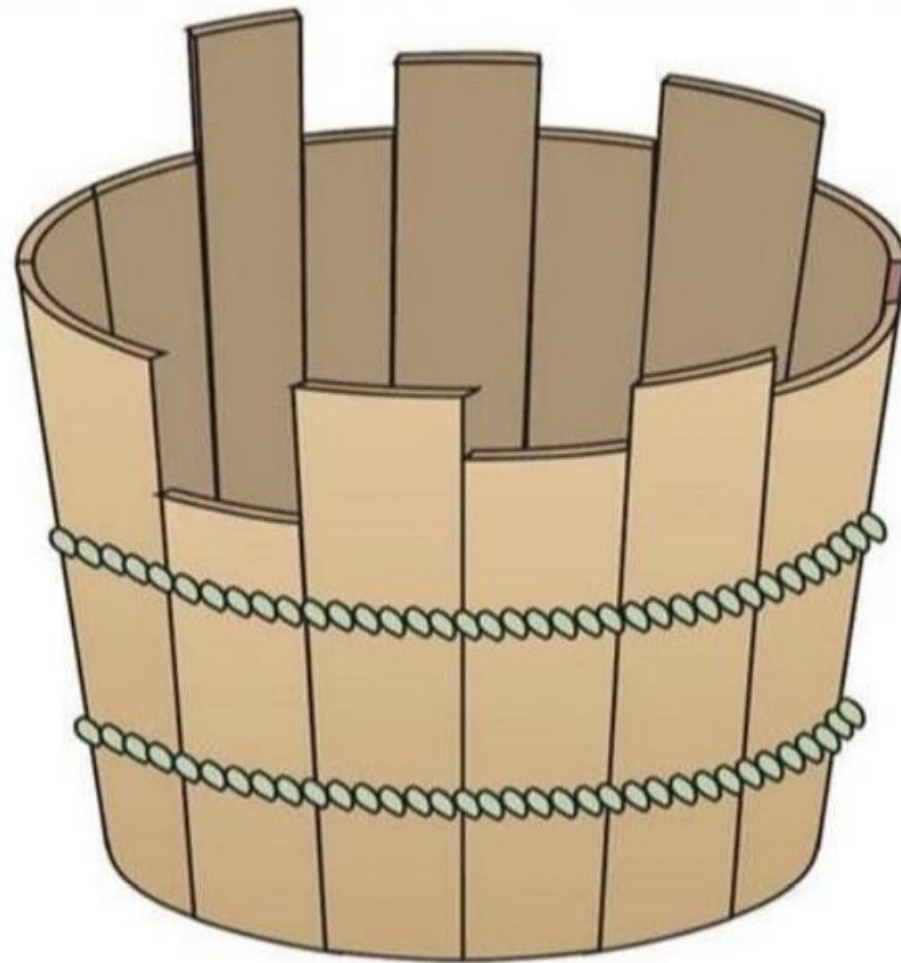
测试点编号	$n$	$k$	$p$	
1	$\leq 2$	$\leq 5$	<div>= 5</div>	
2	$\leq 4$			
3	$\leq 4$			
4	$\leq 10$	$= 1$		
5		$= 10^9$		
6		$\leq 10$		
7				
8	细节决定成败!		$= 40$	
9				
10				
11	$\leq 10^9$	$= 100$		
12		$\leq 250$	$= 200$	
13			$= 300$	
14		$\leq 500$		
15		$\leq 700$		
16		$1000$		
17		$\leq 9500$		
18				



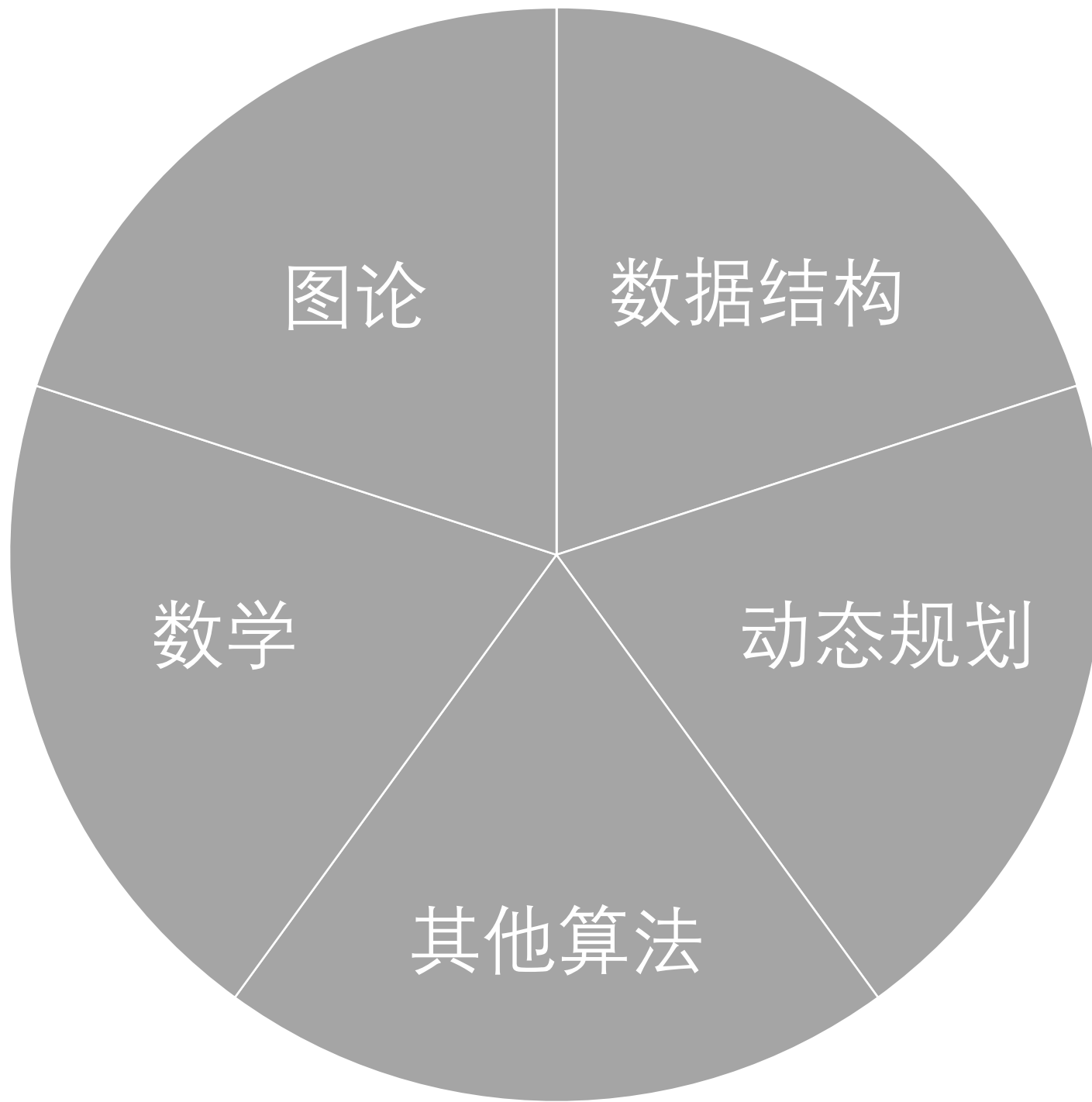
运气?  
这样万事无忧了吗?

# 总结——NOIP到底考什么？

- 算法的熟练掌握
- 思维能力
- 代码能力
- 考场策略安排及细节把控



## Part2 考察算法盘点



# 其他算法——基础算法

- 模拟
- 搜索
- 枚举
- 贪心
- 二分（分治）
- 高精度
- 排序
- .....

# 其他算法——字符串相关

- KMP算法
- 字符串Hash
- 字符串读写!

# 数学

```
graph LR; A[数学] --- B[数论]; A --- C[组合数学]; B --- D[质数]; B --- E[快速幂]; B --- F[GCD与LCM]; B --- G[逆元]; D --- H[判定]; D --- I[筛法]; F --- J[欧几里德算法]; F --- K[扩欧]; G --- L[概念]; G --- M[费马小定理];
```

数论

组合数学

质数

快速幂

GCD与LCM

逆元

判定

筛法

欧几里德算法

扩欧

概念

费马小定理

# 数学

```
graph LR; A[数学] --> B[数论]; A --> C[组合数学]; C --> D[四大计数原理]; C --> E[组合数]; C --> F[常见数列]; D --> G[加法原理]; D --> H[减法原理]; D --> I[乘法原理]; D --> J[除法原理]; F --> K[卡特兰数]; F --> L[错位排列];
```

数论

组合数学

四大计数原理

组合数

常见数列

加法原理

减法原理

乘法原理

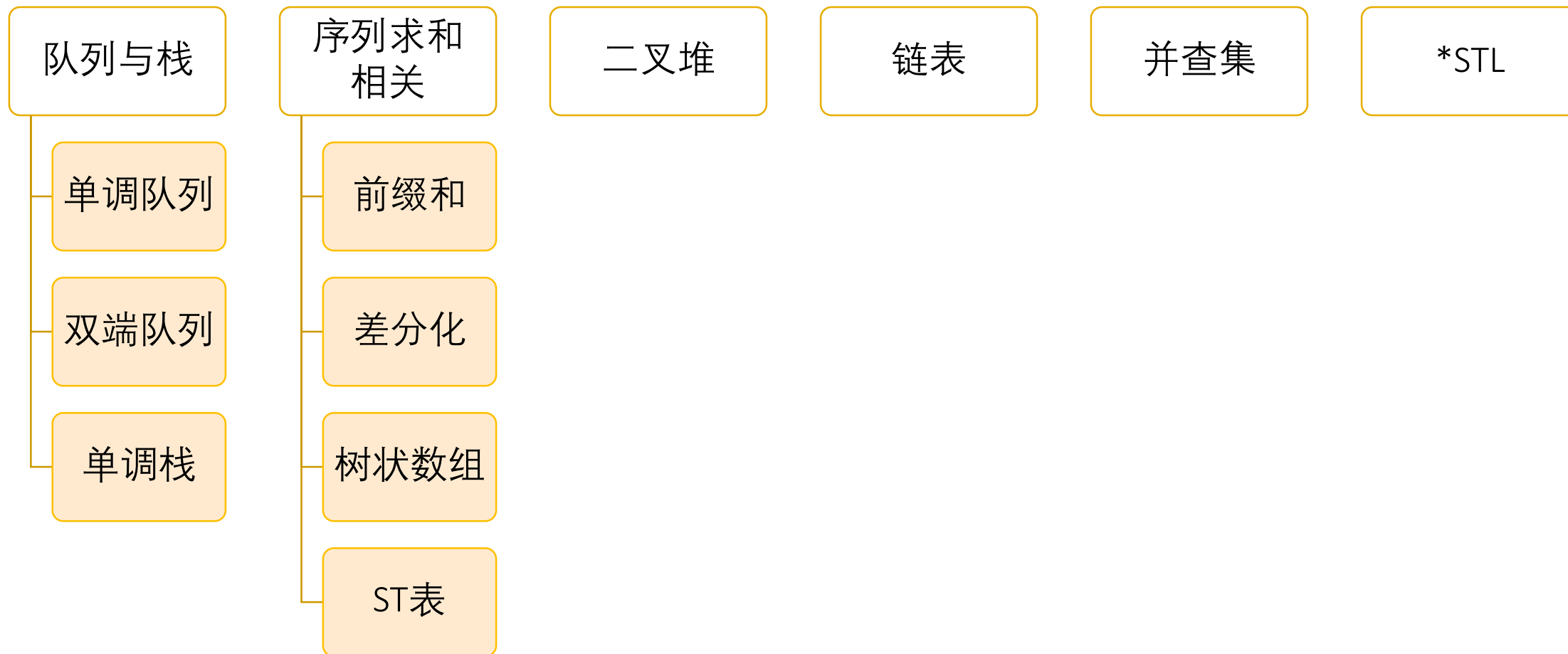
除法原理

卡特兰数

错位排列



# 数据结构



# 图论

## 基础

基本概念

图的存储

DFS与BFS

连通性问题

## 最短 路

无边权图

- BFS

非负权图

- Dijkstra

负权图

- Bellman-Ford

Floyd

- 传递闭包

## Tarjan

强连通分量

割点与点双  
联通分量

割边与边双  
联通分量

## 特殊 的图

二分图

- 判定与匹配

拓扑图  
(DAG)

- 拓扑排序

\*环套树

# 图论——树专题

## 基础

- 树的存储
- 树上遍历基本操作——DFS

## 倍增

- LCA（最近公共祖先）的概念与求法
- 信息维护

## DFS序

- 维护子树信息
- 维护各点到根路径的信息——欧拉序

## 其他

- 最小生成树与Kruskal算法
- 经典问题：直径、重心

# 动态规划（DP）——基础篇

## 你需要了解的基本概念

- 动态规划的本质
- 状态
- 转移

## 经典问题与分类

- 最长公共子序列
- 最长上升子序列
- 各类背包问题
- 区间DP
- 状态压缩DP
- 概率DP

# 动态规划 (DP) ——进阶篇

## 非序列结构DP

- 树形DP
- DAG上DP

## 转移优化

- 决策单调性
- 利用数据结构
- 四边形不等式

# Part3 备考建议&训练指南

```
graph LR; A[明确定位] --> B[自我认知]; B --> C[设计训练]; C --> D[执行计划];
```

明确定位

自我认知

设计训练

执行计划

# 明确定位——你的梦想是什么？





# 针对不同定位的建议

## 冲击省一

- 目标：提高成绩
- 明确取舍
- 高效学习

## 希望在未来进军NOI

- 目标：打好基础
- 切勿急躁!!!

## 今年即向NOI发起冲击

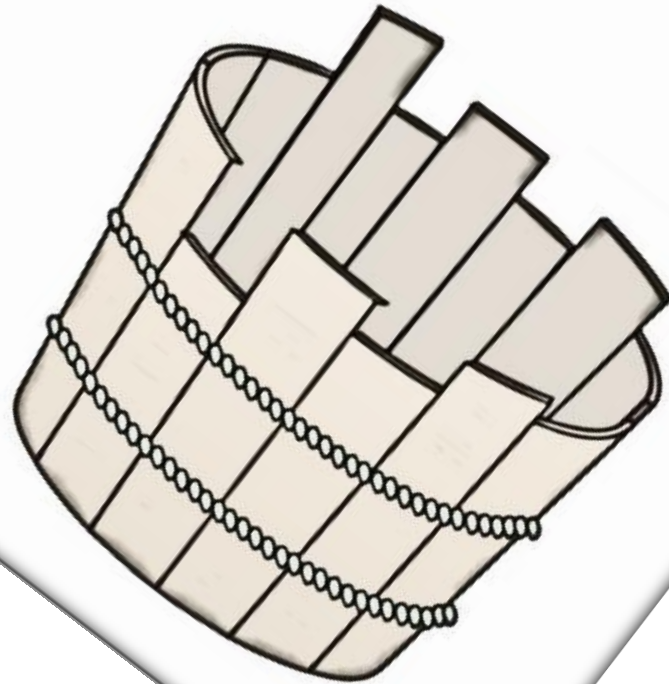
- 对基础查缺补漏
- 适量的超纲内容
- 多训练短板

# 自我认知——发现短板

修补短板是最高效、且必要的

总结——NOIP到底考什么？

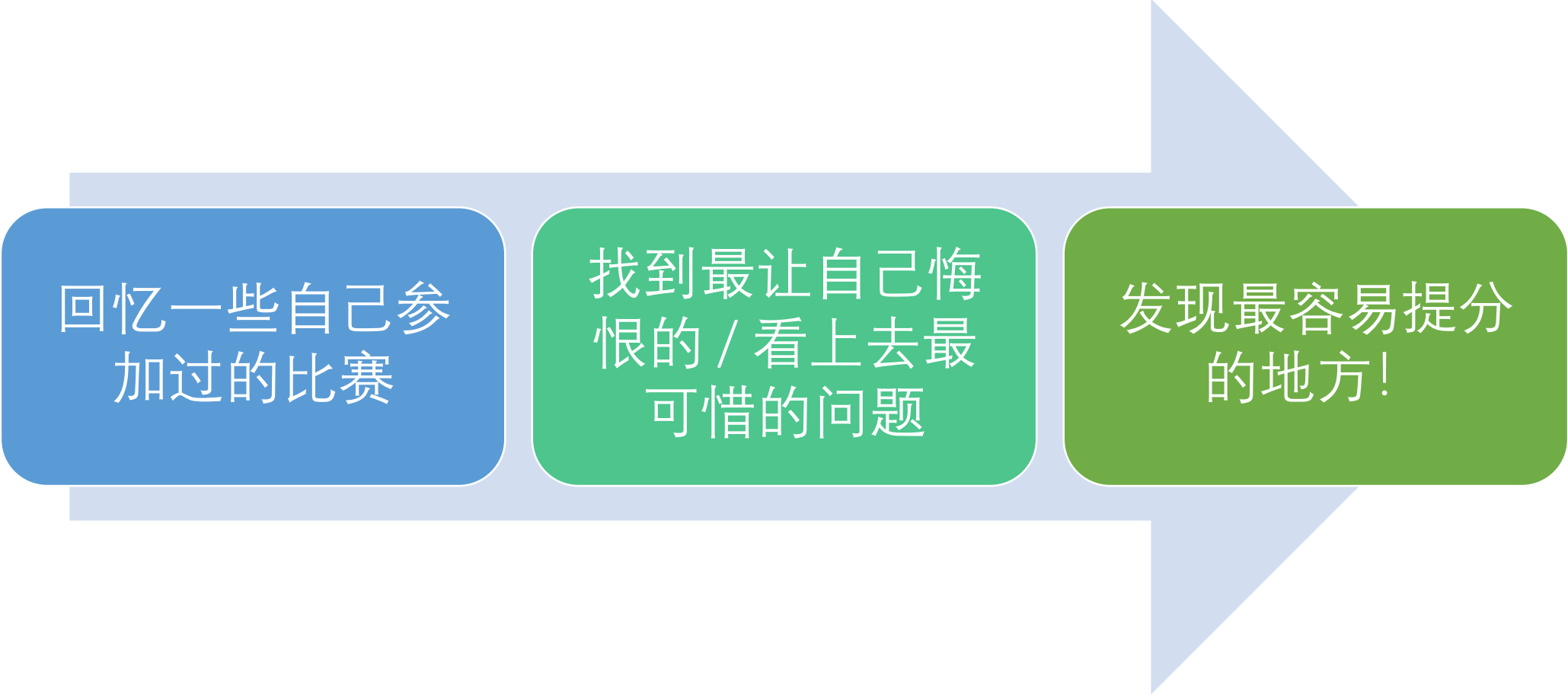
- 算法的熟练掌握
- 思维能力
- 代码能力
- 考场策略安排及细节把控



# 你有没有过下面的这些经历？

- 我要是这个地方没写错我就多100分了！
- 这个算法我明明学过，这个套路也很简单，我为什么没看出来？
- 这个题再仔细想想其实也不难，我为什么就想不到？
- 这个题我明明都想出来了，要是写完了我就多100分了！
- 我要是不做这个题去拿另外两个题的部分分，我就一等奖了！
- 这个算法那么简单，我要是学过我就会做这个题了！
- .....

# 自我认知的最好方法——马后炮



回忆一些自己参加过的比赛

找到最让自己悔恨的 / 看上去最可惜的问题

发现最容易提分的地方！

# 常见训练方法及其针对性

## 书籍或网络资料

- 学习新算法
- 深入理解算法

## 专题刷题

- 深入理解算法
- 训练算法灵活运用

## 试题泛做

- 训练算法灵活运用
- 查缺补漏

## 模拟考试或网络赛

- 训练算法灵活运用
- 考场策略、细节把控

# 如何训练代码能力？

- <https://www.zhihu.com/question/57735917/answer/154155313>
- ↑这个罗哲正在知乎上的回答已经非常完美了。
- ↓总结一下就是
- 固定模板的写法
- think twice, code once
- 多写
- 个人的两个补充↓
  1. 写代码时一定要有意地去抠每个细节，不要寄希望于debug
  2. 平时写代码当成考试，尽可能追求准确率

## Part4 考场技巧&注意事项

# 关于系统



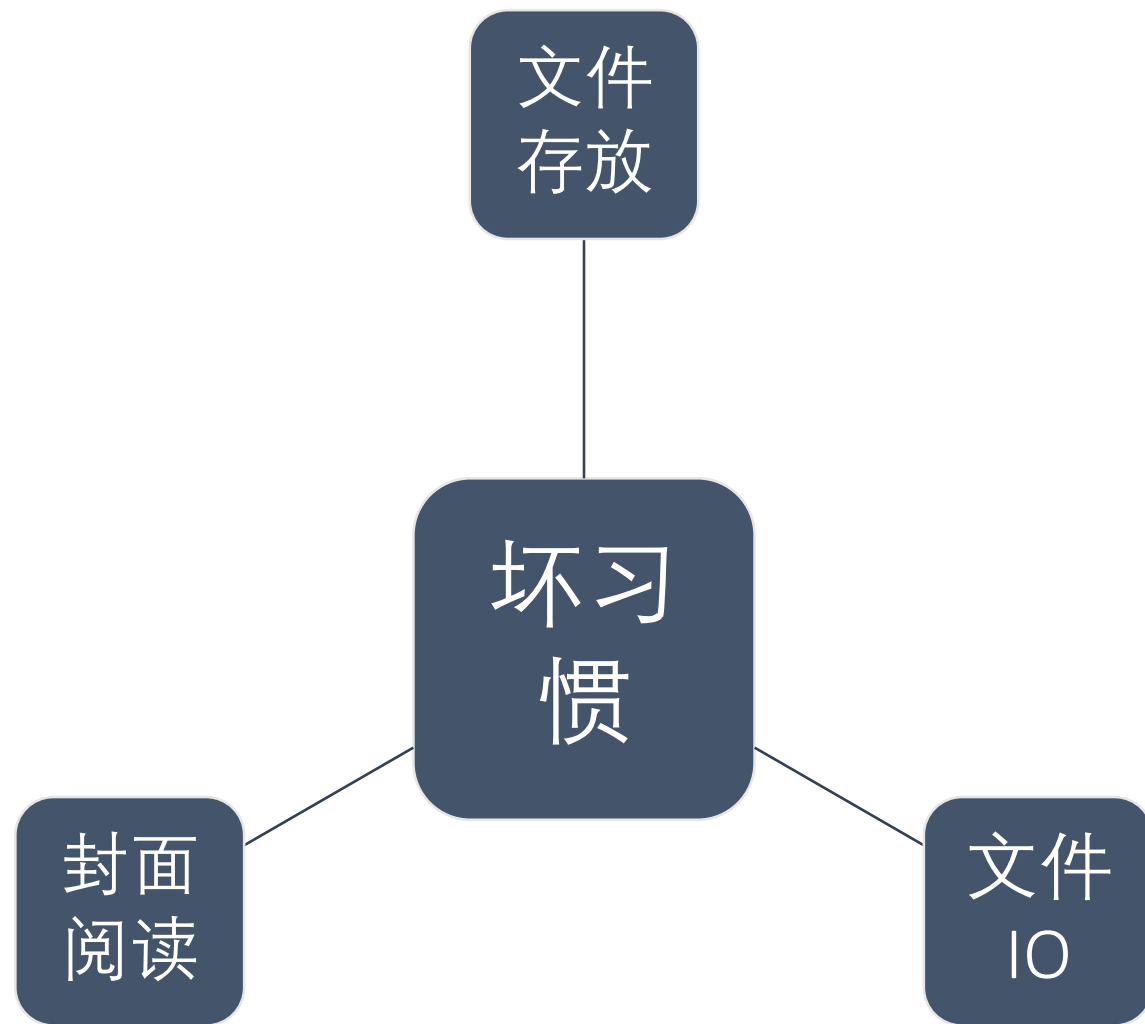
推荐使用

NOI Linux

无论你使用什么

在平时训练中与之保持一致





# 文件存放

- 在桌面工作,
- 考试结束前放到指定位置?
- 直接在指定位置工作!

# 文件IO

```
int main() {  
    // freopen("inverse.in", "r", stdin);  
    // freopen("inverse.out", "w", stdout);  
  
    #ifndef STDIO  
    freopen("name.in", "r", stdin);  
    freopen("name.out", "w", stdout);  
    #endif  
}
```

# 全国青少年信息学奥林匹克竞赛

## CCF NOI 2018

### 试机

时间：2018 年 5 月 6 日 14:30 ~ 17:30

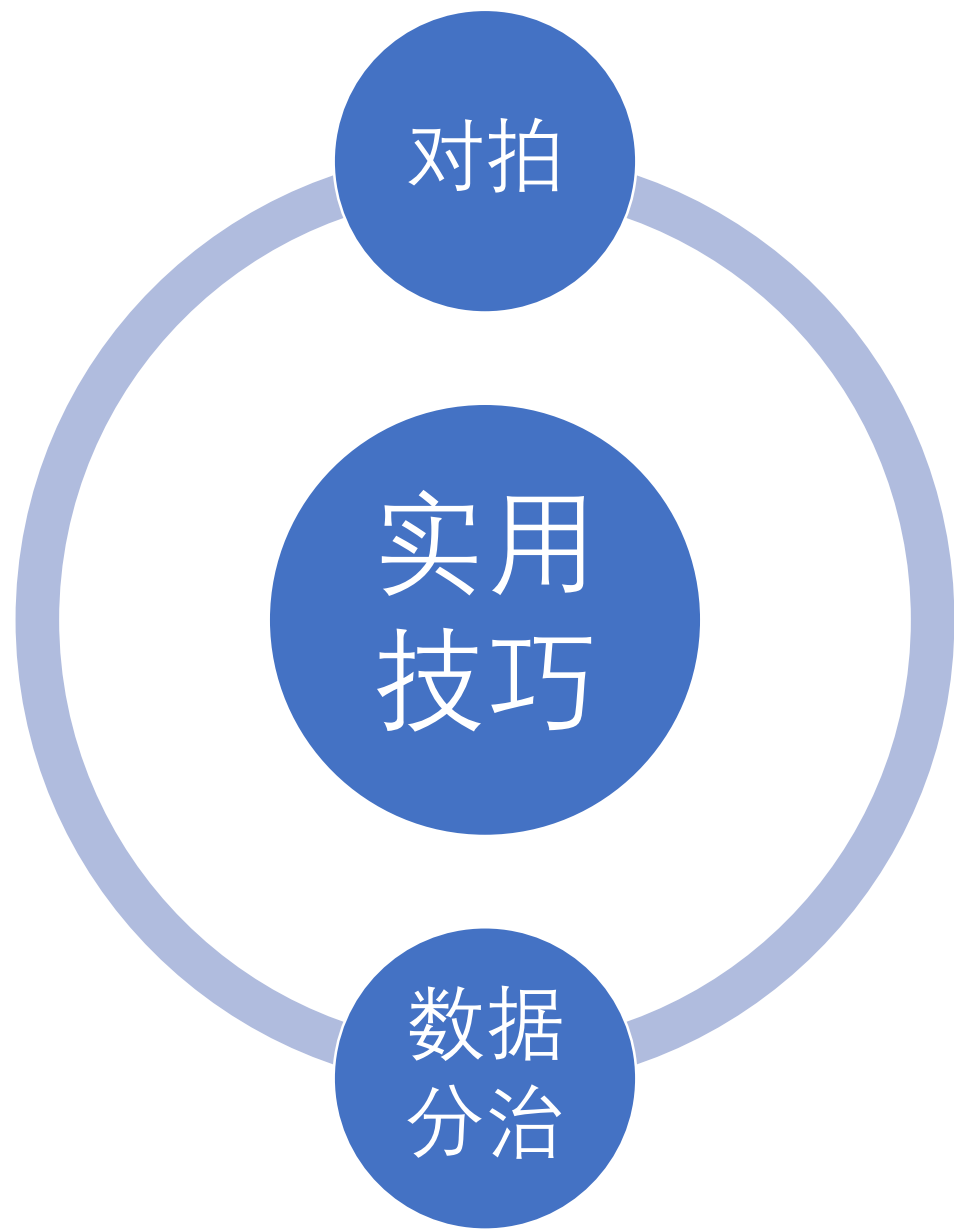
题目名称	一双木棋	IIIDX	假面
题目类型	传统型	传统型	传统型
目录	chess	iiidx	faceless
可执行文件名	chess	iiidx	faceless
输入文件名	chess.in	iiidx.in	faceless.in
输出文件名	chess.out	iiidx.out	faceless.out
每个测试点时限	1.0 秒	2.0 秒	6.0 秒
内存限制	512 MB	512 MB	512 MB
测试点数目	20	20	10
每个测试点分值	5	5	10

提交源程序文件名

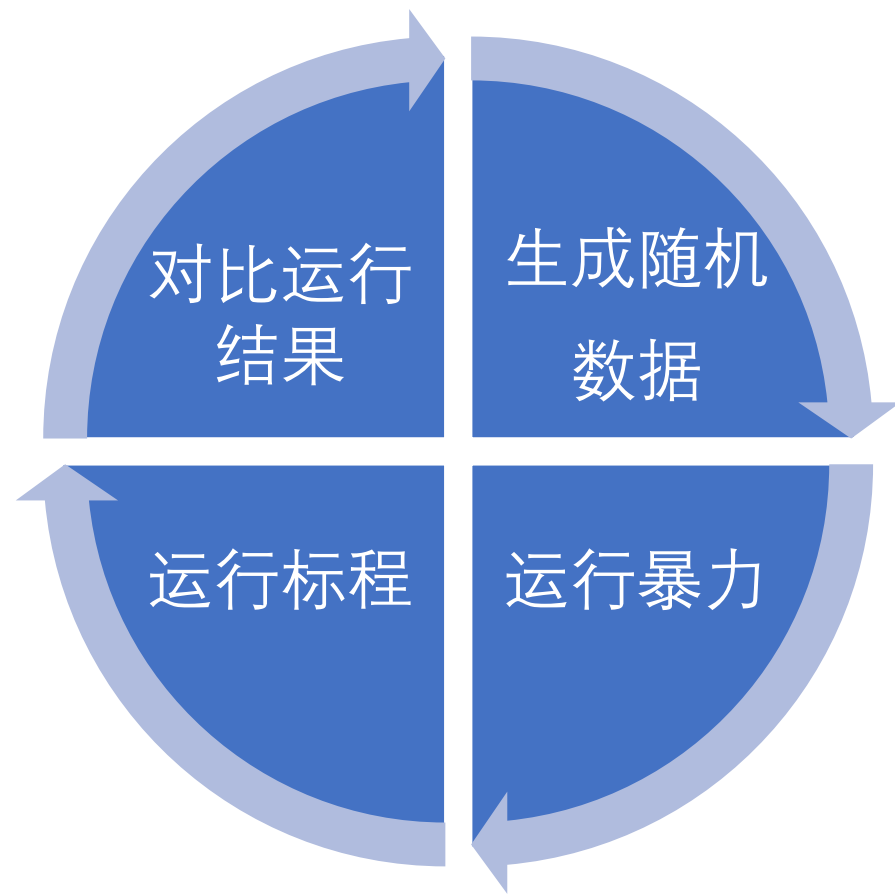
对于 C++ 语言	chess.cpp	iiidx.cpp	faceless.cpp
对于 C 语言	chess.c	iiidx.c	faceless.c
对于 Pascal 语言	chess.pas	iiidx.pas	faceless.pas

编译选项

对于 C++ 语言	-O2 -lm	-O2 -lm	-O2 -lm
对于 C 语言	-O2 -lm	-O2 -lm	-O2 -lm
对于 Pascal 语言	-O2	-O2	-O2



# 对拍



# 要写的代码

暴力

数据生成器

不要忘记srand

建议在脚本里传种子

脚本

```
make
for ((i=1;i<=n;++i))
do
    echo testcase $i:
    ./gen
    echo std running...
    ./std
    echo force running...
    ./force
    if diff name.out name.ans
    then
        echo Accepted!!!!
    else
        echo Wrong Answer....
        break
    fi
done
```

# 数据分治

针对不同性质的测试点使用不同算法

辅助（建议）

使用namespace



# 一些其他注意事项——考试进行时篇

- 开场后先读完所有题和子任务信息
- 仔细审题
- 对于**所有**题目，原则上都不能完全放弃，可以选择做一些性价比高的部分完成

# 一些其他注意事项——收尾篇

- 考试结束前5分钟\*\*一般建议\*\*停止码代码，并做以下检查：
- 检查文件存放位置是否正确，各文件夹名是否正确；
- 检查源程序文件名是否正确；
- 检查输入输出文件名是否正确；
- 检查输出格式是否按照题目要求，不要输出多余信息（特别是调试信息）；
- 检查源程序能否编译通过。
- 考试结束后记得带走准考证

祝大家在NOIP2018获得好成绩

Questions are welcome.