



华中科技大学  
计算机科学与技术学院  
School of Computer Science & Technology, HUST

# 如何在 OI 赛制下 尽可能得分

华中科技大学  
计算机科学与技术学院  
Expect2004  
2022年10月1日



[www.luogu.com.cn](http://www.luogu.com.cn)

## 加入QQ群

如题，请点击课程页面中加群授权码，加入课程QQ群。

进群后，请查看群公告和群文件中的赛前公告，按要求填写问卷，以便全真模拟准考证号分配。

### 相关资源

[课程团队](#)[作业、模拟赛](#)[加群授权码](#)[请在课程通知查看具体加群方式](#)[个人中心](#)[查看已报名课程及订单](#)

# 目录

本节课将以 NOIP 普及组（2018年及之前），CSP – J 为例

# 目录

---

- OI 赛制的特点
- 得分
  - 时间分配
  - 做题策略
- 失分
  - 低级错误
  - Failed System Test
- 心态调整与环境适应
- 骗分技巧

# OI 赛制的特点

## OI 赛制的特点

---

- 限定时间
  - 完成若干道试题
  - 可能有若干大样例
  - 结束一次提交，无即时反馈
- 前两题：模拟，一般不会考察较难的算法 快而稳  
后两题：基础算法，如动态规划、贪心、数据结构  
2021年第三题（网络连接）考察模拟
- 考虑清楚：时限、细节

# 得分因素

# 时间分配

CSP - J 组考试时间为 8:30-12:00 共3.5小时

时间分配没有标准答案

{  
读题  
思考  
写代码  
检验

沉没成本

得分效率

必须读完所有题！不要犹豫

降低浪费（样例？） 得分效率：先做稳拿的（部分）分，后做难拿的分  
对于新想出来的算法，先验证再打代码



## 得分策略

---

简单题拿满分

困难题部分分

尤其重视暴力分 (dfs,  $n^x$  枚举) 【部分分】

# 为什么要打暴力分？

打比赛不一定 AC 的题足够多才能拿一等奖。打到的分数能过线就能拿一等奖。

以 NOIP 2021 为例：

- 浙江省的一等奖分数线为 186 分。
- 江苏省的一等奖分数线为 150 分。
- 山东省的一等奖分数线为 120 分。

# 为什么要打暴力分？

大家签到题都没有问题，100 分。

对于剩下来的三道题，最低一档数据范围分值：

- 第二题数列，最低档有 4 个点，每个点 5 分，共有  $4 \times 5 = 20$  分。
- 第三题方差，最低档有 3 个点，每个点 4 分，共有  $3 \times 4 = 12$  分。
- 第四题棋局，最低档有 2 个点，每个点 4 分，共有  $2 \times 4 = 8$  分。

加起来共  $100 + 20 + 12 + 8 = 140$  分。到分数线已经很近了。再加上别的部分分，混个一等奖显然没问题。

# 为什么要打暴力分？

---

此外，打暴力有个很重要的作用，就是对拍。

OI 赛制下，出题人提供的仅有几个样例，而这样例也可能出现**强度过低**的情况，于是将你的程序与暴力程序对拍可以更好发现错误。

对拍在 Windows 下的实现

## 部分分与性质挖掘

- $N$  很小 打表 穷举 手玩
- $N$  20 左右  $2^N$  搜索（也有可能是状压正解）
- $N$  500 左右  $N^3$  暴力
- $N$  1000 – 5000  $N^2$  暴力（不一定）
- 正解可能是由劣解**步步优化**而来。根据优化的层次，出题人会设置梯度。
- 特殊性质

# 特殊性质

值域的特殊限制

图形态的特殊限制

只要求解决子问题

其他特殊性质

如图退化为树

只有两层的树

树退化为一链；树是菊花图



每个节点，出度和入度都为 1，链头入读为 0，链为出度为 0。

矛盾的特殊性与普遍性

特殊性质是问题的特化，引导正解

# 如何写多档部分分？

---

namespace 在部分分中的应用

namespace: 命名空间，不同命名空间可以变量重名。

解决多份代码，变量、数组重名的问题

```
#include<bits/stdc++.h>
using namespace std;
// 这里放各种要输入的东西的定义
namespace Reader{
    void read(){ ... } // 这里读入所有数据
}
namespace Sub1{           // 第一种数据范围的做法
    // 这里放这种做法其他的变量 / 函数
    void mian(){ ... }
}
namespace Sub2{           // 第二种数据范围的做法
}
...
int main(){
    Reader :: read();           // 先读入所有东西
    if( ... ) Sub1 :: mian(); else // 对于不同范围采用不同做法
    if( ... ) Sub2 :: mian(); else ...
}
```



## 设法区别出不同的数据范围

有些题目比较良心，第一行会输入该测试点的编号。或者它会有很明显的特征，比如：

注意表头

测试点编号	$n$ 	特殊性质
1 ~ 2	7	无
3 ~ 5	199	无
6 ~ 8	1999	无
9 ~ 11	49991	A
12 ~ 15	262143	B
16	99995	无
17 ~ 18	199995	无
19 ~ 20	299995	无

# 设法区别出不同的数据范围

再如 NOIP2016 D1T2 天天爱跑步

测试点编号	$n$	$m$	约定
1	= 991	= 991	所有人的起点等于自己的终点， 即 $S_i = T_i$
2			
3	= 992	= 992	$W_j = 0$
4			
5	= 993	= 993	无
6	= 99994	= 99994	树退化成一条链，其中1与2有边， 2与3有边， $\dots$ ， $n-1$ 与 $n$ 有边
7			
8			
9	= 99995	= 99995	所有的 $S_i = 1$
10			
11			
12			
13	= 99996	= 99996	所有的 $T_i = 1$
14			
15			
16			
17	= 99997	= 99997	无
18			
19			
20	= 299998	= 299998	

失分

## 失分绪论

---

不按规定建立文件目录

文件命名错误

文件读写操作错误

备份：代码、试题

本机 AC 提交 CE (Linux or Windows)

写挂了 (最常见)

## 提交环节

---

留出时间检查提交目录、文件命名、文件操作。

# 提交 CE 问题

---

NOI Linux 下的编译

# 写挂

---

情况一

部分分叠加策略

情况二

对拍

Corner Cases 测试

小数据（极限数据）的构造

# 容易导致挂分的 Trap Point

## 一 编译

- `x0, x1, y0, y1, unix`  
保留字，禁止使用

## 二 一些常量

- `MAXN` 多大？`N` 还是 `2N`？
- 预处理赋的 `INF` 至少多大？
- 二分 `R = 2e9; int ans = R, mid = (R + R) / 2;`  
爆 `int` 需开 `long long`



# 容易导致挂分的 Trap Point

---

## 三 变量类型

- $\text{int} * \text{int}$  很可能爆  $\text{int}$  (NOIP2018龙虎斗)
- 答案最大多大?

## 四 多测

- 数据一定要读完
- 多测清空问题

# 心态与环境

# 心态与环境

---

环境的影响：

- 键盘不顺手
- 集成开发环境不习惯
- Dev-C++ 动态调试寄了

# 一些乱搞

# 模拟退火 \*

如果没看出性质，还有一个可能的路径。如果一次操作使得答案变得更小，那就保留这次操作；否则有概率接受这次操作。

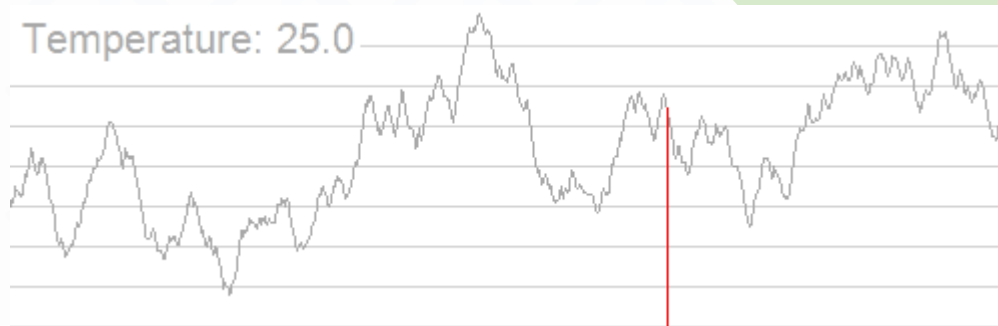
这可以用模拟退火实现。具体而言，

- 初始时令  $T = T_{\max}$ 。接着进入循环：
- 生成一个新解（随机找一个位置操作）。若新解更优就接受，否则以  $\exp(-\frac{ans - newans}{rT})$  的概率接受。
- 令  $T$  乘上  $\Delta$ 。
- 若  $T < T_0$  就退出循环，否则继续这个过程。

其中  $T_{\max}, T_0, \Delta, r$  都是你自己决定的常数。需要进行调参。

# 模拟退火

---



## 多重贪心 \*

---

对于一个题目，你有若干 **错误的** 贪心策略（一定概率正确）。

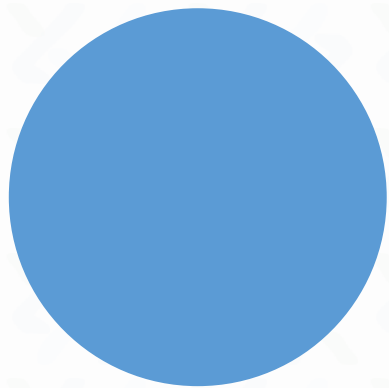
每一个贪心策略，得到的答案都比最终答案 大/小。  
都跑一遍，取答案  $\max/\min$ 。

Q & A





华中科技大学  
计算机科学与技术学院  
School of Computer Science & Technology, HUST



谢谢观看



[www.luogu.com.cn](http://www.luogu.com.cn)