

考试策略

zhou888

雅礼中学

November 1, 2018

Preface

考试策略在每次考试中都是非常重要的

Preface

考试策略在每次考试中都是非常重要的

然而我并不会

Preface

考试策略在每次考试中都是非常重要的

然而我并不会

既然我不会，我们就先**随机**请一些平时考试成绩特别好的同学上来分享一下自己的考试经验

X503-E04

X503-E04

X503-C06

X503-E04

X503-C06

X503-D06

X503-E04

X503-C06

X503-D06

X503-D02

X503-E04

X503-C06

X503-D06

X503-D02

还有考试的时候从来没停下手指的X503-E05

考前背好vimrc

考前背好vimrc

不要用guide编程

先把三道题看完，不要急着写，确定自己的切题顺序，大概看一眼每一题的暴力分。

先把三道题看完，不要急着写，确定自己的切题顺序，大概看一眼每一题的暴力分。

一定要把题看清楚，如果由较小的样例或样例解释一定要手算一下。

先把三道题看完，不要急着写，确定自己的切题顺序，大概看一眼每一题的暴力分。

一定要把题看清楚，如果由较小的样例或样例解释一定要手算一下。

题目里的数据一定不要错过（不要把 $1e8+7$ 看成 $1e9+7$ ）。

先把三道题看完，不要急着写，确定自己的切题顺序，大概看一眼每一题的暴力分。

一定要把题看清楚，如果由较小的样例或样例解释一定要手算一下。

题目里的数据一定不要看错（不要把 $1e8+7$ 看成 $1e9+7$ ）。

虽然NOIP的题目顺序与难度大抵是差不多的，但题目难度对于不同的人是不同的，即使是NOIP，题目顺序不一定是难度顺序。



先把三道题看完，不要急着写，确定自己的切题顺序，大概看一眼每一题的暴力分。

一定要把题看清楚，如果由较小的样例或样例解释一定要手算一下。

题目里的数据一定不要看错（不要把 $1e8+7$ 看成 $1e9+7$ ）。

虽然NOIP的题目顺序与难度大抵是差不多的，但题目难度对于不同的人是不同的，即使是NOIP，题目顺序不一定是难度顺序。

例如NOIP2016最难的题是D1T2，NOIP2017最简单的题是D1T2

对于暴力比较好写或者感觉比较难的题，一定要把暴力先写好（方便对拍且防止没分）



对于暴力比较好写或者感觉比较难的题，一定要把暴力先写好（方便对拍且防止没分）

NOIP中一般有送分题，对于这种题一定要拿稳。

对于暴力比较好写或者感觉比较难的题，一定要把暴力先写好（方便对拍且防止没分）

NOIP中一般有送分题，对于这种题一定要拿稳。

考虑到noip的样例强度，对拍比较好写的题一定要对拍。即使无法对拍，也要自己造一些数据手算一下。

对于暴力比较好写或者感觉比较难的题，一定要把暴力先写好（方便对拍且防止没分）

NOIP中一般有送分题，对于这种题一定要拿稳。

考虑到noip的样例强度，对拍比较好写的题一定要对拍。即使无法对拍，也要自己造一些数据手算一下。

考虑到noip的数据强度，暴力也尽量写复杂度较优的或者不好把你的暴力卡到极限的（可以用random_shuffle函数）。

当你想了很久想不出来一道题或者想出一道题时，再把题目看一遍。

当你想了很久想不出来一道题或者想出一道题时，再把题目看一遍。

如果在研究一道题正解没有什么思路的话，可以先研究部分分，一般部分分都是对正解有提示作用的。

当你想了很久想不出来一道题或者想出一道题时，再把题目看一遍。

如果在研究一道题正解没有什么思路的话，可以先研究部分分，一般部分分都是对正解有提示作用的。

对于一些感觉比较偏结论的题，可以考虑找找规律

当你想了很久想不出来一道题或者想出一道题时，再把题目看一遍。

如果在研究一道题正解没有什么思路的话，可以先研究部分分，一般部分分都是对正解有提示作用的。

对于一些感觉比较偏结论的题，可以考虑找找规律

如果你觉得这道题正解细节较多或比较难写，一定要先去把其他题能拿的分和这题的暴力写了。

当你想了很久想不出来一道题或者想出一道题时，再把题目看一遍。

如果在研究一道题正解没有什么思路的话，可以先研究部分分，一般部分分都是对正解有提示作用的。

对于一些感觉比较偏结论的题，可以考虑找找规律

如果你觉得这道题正解细节较多或比较难写，一定要先去把其他题能拿的分和这题的暴力写了。

永远不要在还有其他题可写的情况下去头铁写一道你不一定能搞出来的题！！！！

123 Pairs

有 $2n$ 个数字, $1 \rightarrow 2n$. 你要将他们分成 n 个Pair, 使得每个数字恰好在一个Pair中. 要求如果 i, j 在一个Pair中, 那么 $|i - j| \leq 3$. 并且恰好有 A 个Pair满足 $|i - j| = 1$, 恰好有 B 个Pair满足 $|i - j| = 2$, 恰好有 C 个Pair满足 $|i - j| = 3$. 计算有多少种不同的方案, 对 $10^9 + 7$ 取模.

$$n, A, B, C \leq 5000, A + B + C = n$$

部分分 $C = 0$

Solution of $C=0$

用用 $L1$ 表示标号差为1的连接, $L2, L3$ 类似.

Solution of $C=0$

用用 $L1$ 表示标号差为1的连接, $L2$, $L3$ 类似.

考虑 1 与谁相连

Solution of $C=0$

用用 $L1$ 表示标号差为1的连接, $L2, L3$ 类似.

考虑 1 与谁相连

如果 1-2 相连, 就变成 $n-1$ 对点的子问题.

Solution of $C=0$

用用 $L1$ 表示标号差为1的连接, $L2, L3$ 类似.

考虑 1 与谁相连

如果 $1-2$ 相连, 就变成 $n-1$ 对点的子问题.

如果 $1-3$ 相连, 那么 $2-4$ 必须连, 就变成 $n-2$ 对点的子问题.

Solution of $C=0$

用用 $L1$ 表示标号差为1的连接, $L2, L3$ 类似.

考虑 1 与谁相连

如果 $1-2$ 相连, 就变成 $n-1$ 对点的子问题.

如果 $1-3$ 相连, 那么 $2-4$ 必须连, 就变成 $n-2$ 对点的子问题.

也就是每次消耗 $L1 \times 1$ 或者 $L2 \times 2$.

Solution

考虑第一个位置的连接情况.

Solution

考虑第一个位置的连接情况.

考虑怎么连接才能填满一个前缀.

Solution

考虑第一个位置的连接情况.

考虑怎么连接才能填满一个前缀.

$$L1 \times 1 : 1 - 2$$

$$L3 \times 1 + L1 \times 1 : 1 - 4, 2 - 3$$

$$L3 \times 3 : 1 - 4, 2 - 5, 3 - 6$$

$$L2 \times 2 + L3 \times v (v \geq 0) : 1 - 3, 2 - 5, 4 - 7 \dots, 2v - (2v + 2)$$

Solution

设这 4 种各消耗了 x_1, x_2, x_3, x_4 个。

并设第 4 种中 v 的和是 S .

枚举 x_1, x_3

那么 $x_2 = A - x_1, x_4 = B/2, S = C - x_3 \times 3 - x_2$

方案数即为 $Ans = \frac{(x_1 + x_2 + x_3 + x_4)! \times \binom{x_4 + S - 1}{x_4 - 1}}{x_1! x_2! x_3! x_4!}$.

如果你实在是没有正解思路或者你发现一种很好的乱搞方法的话，可以考虑乱搞。

如果你实在是没有正解思路或者你发现一种很好的乱搞方法的话，可以考虑乱搞。

你要想出题人一般想不到你的乱搞是怎么写的。

如果你实在是没有正解思路或者你发现一种很好的乱搞方法的话，可以考虑乱搞。

你要想出题人一般想不到你的乱搞是怎么写的。

尽量写正确率高不好想到的乱搞。

如果你实在是没有正解思路或者你发现一种很好的乱搞方法的话，可以考虑乱搞。

你要想出题人一般想不到你的乱搞是怎么写的。

尽量写正确率高不好想到的乱搞。

拿暴力套乱搞

如果你实在是没有正解思路或者你发现一种很好的乱搞方法的话，可以考虑乱搞。

你要想出题人一般想不到你的乱搞是怎么写的。

尽量写正确率高不好想到的乱搞。

拿暴力套乱搞

最优化题目可以取多个乱搞嵌套取最优值。

写程序之前先分析一遍大概的过程，把尽可能多的细节考虑清楚。

写程序之前先分析一遍大概的过程，把尽可能多的细节考虑清楚。

没有考虑清楚的话只会增加调试时间，越写越难写

写程序之前先分析一遍大概的过程，把尽可能多的细节考虑清楚。

没有考虑清楚的话只会增加调试时间，越写越难写

如果代码难度较大，先想一想怎么写好写。

写程序之前先分析一遍大概的过程，把尽可能多的细节考虑清楚。

没有考虑清楚的话只会增加调试时间，越写越难写

如果代码难度较大，先想一想怎么写好写。

尽量减少代码常数，可能可以取得意想不到的分数（具体可以询问X503-C07)

编译前先认真检查一下，静态查一下错。

编译前先认真检查一下，静态查一下错。

测试时按照要求编译命令（关 -O2 -std=c++11）

编译前先认真检查一下，静态查一下错。

测试时按照要求编译命令（关 -O2 -std=c++11）

编译时加上 $-Wall$ $-Wextra$ $-Wshadow$ 命令减少出错概率

编译前先认真检查一下，静态查一下错。

测试时按照要求编译命令（关 `-O2 -std=c++11`）

编译时加上 `-Wall -Wextra -Wshadow` 命令减少出错概率

RE时试一下 `-fsanitize = address -fsanitize = undefined` 命令

编译前先认真检查一下，静态查一下错。

测试时按照要求编译命令（关 `-O2 -std=c++11`）

编译时加上 `-Wall -Wextra -Wshadow` 命令减少出错概率

RE时试一下 `-fsanitize = address -fsanitize = undefined` 命令

样例过了还是多对拍。

编译前先认真检查一下，静态查一下错。

测试时按照要求编译命令（关 `-O2 -std=c++11`）

编译时加上 `-Wall -Wextra -Wshadow` 命令减少出错概率

RE时试一下 `-fsanitize = address -fsanitize = undefined` 命令

样例过了还是多对拍。

无论对拍还是没对拍，都要造**极限**数据加上上面那个命令测试有无数组越界和爆int/long long情况

NOIP当正常的考试来考，不要太紧张

NOIP当正常的考试来考，不要太紧张

不要受到周围同学的影响

NOIP当正常的考试来考，不要太紧张

不要受到周围同学的影响

考试时一定要专注，在想某道题时就专注于那道题，不要想一些其他的事情。

NOIP当正常的考试来考，不要太紧张

不要受到周围同学的影响

考试时一定要专注，在想某道题时就专注于那道题，不要想一些其他的事情。

及时放弃，如果思路很乱，先把这题放在一边,去搞好搞的题

NOIP当正常的考试来考，不要太紧张

不要受到周围同学的影响

考试时一定要专注，在想某道题时就专注于那道题，不要想一些其他的事情。

及时放弃，如果思路很乱，先把这题放在一边,去搞好搞的题

如果感觉状态不太好，可以去洗一下脸

考试结束之前不要再打题，再检查一遍数组（不要开大/小）。

考试结束之前不要再打题，再检查一遍数组（不要开大/小）。

再编译一遍测样例，检查有没有注释没删,测样例一定要用diff

考试结束之前不要再打题，再检查一遍数组（不要开大/小）。

再编译一遍测样例，检查有没有注释没删,测样例一定要用diff

建好子文件夹，检查文件名以及freopen

考试结束之前不要再打题，再检查一遍数组（不要开大/小）。

再编译一遍测样例，检查有没有注释没删,测样例一定要用diff

建好子文件夹，检查文件名以及freopen

检查数据范围(读入long long,模数，取模不要爆负)

考试结束之前不要再打题，再检查一遍数组（不要开大/小）。

再编译一遍测样例，检查有没有注释没删,测样例一定要用diff

建好子文件夹，检查文件名以及freopen

检查数据范围(读入long long,模数，取模不要爆负)

多组数据一定要多复制几遍测，记得数组清零,注意不要用y1等变量

noi系列大部分的考试都是分Day1/Day2的，在NOIP Day1之后不要去关心第一天的成绩

noi系列大部分的考试都是分Day1/Day2的，在NOIP Day1之后不要去关心第一天的成绩

如果Day1没考好，要坚信自己能翻回来。

noi系列大部分的考试都是分Day1/Day2的，在NOIP Day1之后不要去关心第一天的成绩

如果Day1没考好，要坚信自己能翻回来。

如果觉得自己Day1还可以，Day2更应该调整心态，稳扎稳打

考试时你要相信题目noip不会很难。

考试时你要相信题目noip不会很难。

最后以去年某位同学的成绩来激励大家：

math	complexity	park	cheese	treasure	phalanx
5	10	80	100	80	100

考试时你要相信题目noip不会很难。

最后以去年某位同学的成绩来激励大家：

math	complexity	park	cheese	treasure	phalanx
5	10	80	100	80	100

希望大家在NOIP中不要挂分，取得好成绩。

Thanks