2020/05/25--2020/05/31 周报

团队训练

5.30一次团体赛（记录下周再完善）

Codeforces Round 638 Div. 2

题目	A	B	C	D	E	F	G	H
通过	√	√	√	√				
补题								

个人训练 - Great_designer

比赛

六月底烤漆结束前基本不会有（除非被拉去团体赛）

5.30参加一次团体赛

学习总结

本周完成了number_theory_sqrt_decomposition。感谢原作者队伍提供的题目。

OJ计划更新完了前200题：GitHub BUAAOJ计划

本周推荐

板子分析系列推完了，推一道中等题好了OJ编号147：贪心方法与优先队列

个人训练 - serein

比赛

2020.05.24 Codeforces Round #644 (Div. 3)

题目	A	B	C	D	E	F	G	H
通过	√	√	√					
补题				√	√	√	√	√

学习总结

周一用了一个下午把周日晚上的div3的题都补了cf的题的特点或者说这类题的特点感觉就是题面比较长，背景也比较丰富。
刚开始做英语题面的题，速度可能比较慢，有的时候读题是要花一些时间。当过了读题的坎儿后，接下来的过程就是抽象出问题考察的数学问题或者模型，这个过程有时也会比较难，有些题目当你抽象出一个比较好的数学模型时，写起来会简单很多。经过这几周的题目训练，读题应该基本没有问题了。
我感觉在过了入门的阶段之后，接下来要做的应该是学习更多知识点，更好的进行题目训练和比赛训练。

本周推荐

Codeforces Round #644 (Div. 3)

E

个人训练 - kongyou

比赛

参加团体赛，完成了A B题 其中B题采用了暴力的解法，正在用更好的解法进行补题

学习总结

linux环境下的程序运行

我们可以用这样的方法进行程序运行: 1. 选定文件夹, 假设我们的代码是main.cpp, 那么我们可以用这样的方法去编译程序 gcc -o main main.cpp -Wall -lm 在linux环境中我们可以使用ide也可以使用VIM等来写代码

2. 执行代码 如果我们一切顺利，我们在文件夹中输入ls指令能够看到有一个main的文件，这时我们执行./main命令，就可以以命令行的方式输入输出了

linux的常见命令

ls 查看文件夹内容

mkdir 新建文件夹

rmdir 删除文件夹

cd 进入文件夹

cp 复制文件

mv 剪贴文件

vim 启动vim

动态规划

将递归方法重新写成非递归方法，并将子问题的答案系统的记录在一个表内的方法叫动态规划

举一个例子，下面一对儿算法中，由于递归实现产生了过多的重复运算，所以效率不高；通过动态规划，我们减少了重复计算，复杂度也降低为O(n)

```
int Fib(int N) {
    if (N <= 1)   return 1;
    else return Fib(N-1) + Fib(N-2);
}
```

```
int Fibonacci(int N) {
    int i, Last,  NextToLast, Answer;
    if (N <= 1) return 1;
    Last = NextToLast = 1;
    for (i = 2; i < N; i++) {
        Answer = Last + NextToLast;
        NextToLast = Last;
        Last = Answer;
    }
    return Answer;
}
```

本周推荐

一个是gcc的参数介绍https://www.runoob.com/w3cnote/gcc-parameter-detail.html

还有一个是linux的命令https://www.runoob.com/linux/linux-command-manual.html

熟悉linux环境并采用gcc进行编译，能够在最大限度上贴近实际评测环境，从而避免出现由于环境不一致导致的WA

From:
https://wiki.cvbbacm.com/ - **CVBB ACM Team**

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:namespace:week_summary_4&rev=1593582913

Last update: **2020/07/01 13:55**