

# 2020/05/18--2020/05/24 周报

## 团队训练

無。



## 团队会议

## 个人训练 - Great\_designer

### 比赛


无。

### 学习总结

在马先生的基础上施工了[广度优先搜索\\_bfs\\_与标数最短路\\_dijkstra](#)，完善了[整数分拆问题](#)□

### 本周推荐

只好再搬一篇过去的解析：[数论函数和线性筛算法](#)



## 个人训练 - serein

### 比赛

#### 2020.05.21 Codeforces Round #643 (Div. 2)

过题□A□B□D

## 2020.05.22 Accoding 期末保温赛

过题□AK

## 2020.05.24 Codeforces Round #644 (Div. 3)

过题□A□B□C

### 学习总结

这周做Accoding的时候翻了车，一道很简单的括号匹配居然Debug了一个小时，原因是我不会标记栈顶的位置，后来发现用一个散列表就可以动态的标记栈顶的位置了，做题还是不能太着急、大意□643div.2的前四道题做了快两个小时□C题找三角形的匹配用的是前缀和，这个之前没学过所以不会，看答案还是有点懵□644div.3一个半小时做了五道题□D题的思路很简单，但是我T了两次，实在是不会优化了，粘了一个快速判定质数的板子还是T了，E题是一个dfs□但是在Debug中。E题做出来了，在比赛结束5分钟后。看来还是不够熟练。

### 本周推荐

就本周我做的题来看，我会推荐643div.2的C题，技巧是前缀和。

## 个人训练 - kongyou

### 比赛

无（这周考试去了）

### 学习总结

这周主要巩固了离散的基础知识，通过OO课程复习了图的一些算法。 [图论相关](#)


### 本周推荐

在这里我推荐我从OO课程中学习到的编程技巧：离散化与缓存。

离散化是把无限空间中有限的个体映射到有限的空间中去，以此提高算法的时空效率。可以适用于解决只关注元素之间的大小关系而不关注元素本身的值的有关问题。具体的介绍可以看这篇博客 [离散化](#)

而缓存常常适用于我们经常对部分重复情况进行访问的时候。比如对于图的最短路径，我们可以用缓存来进行计算，这可以在很大程度上减少重复计算量，从而提高效率避免TLE□

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

Permanent link:  
[https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:namespace:week\\_summary\\_3&rev=1590336858](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:namespace:week_summary_3&rev=1590336858) 

Last update: **2020/05/25 00:14**