

# Update on Wiki

- 创建了本周训练周报
- 创建了暑期自己第二次加训界面

## 团队训练

[The 2015 ACM-ICPC Asia Beijing Regional Contest](#)

## 每周推荐

**fyh:**

题目大意：一个序列，你可以每次把这个序列前五个重排，然后如果重排后前三个数相同得一分，无论得不得分都会把这三个数删掉，特殊的是如果最后剩下三个数还相同的话也得一分，求最大得分

**tag:**DP DP优化

做法:首先如果当前位置三个都相同我们一定要得上这一分（根据荒谬的方法可以证明（误））

设 $dp[i][x][y]$ 表示当前在第 $i$ 轮，这两个数是 $x,y$ 的最大得分。

分几种转移，一种是三个数都相同的，那么对于 $dp[i-1][][[]]$ 的所有合法状态都应该+1

第二种是有两个数相同的，设相同的数为 $p$ ，另外一个为 $q$ ，那么 $dp[i][q][k]$ 可以从 $dp[i-1][k][p]$ （或者反过来）转移过来。

第三种是很随意的，设那三个数分别为 $p,q,r$ ，那么 $dp[i][q][r]$ 可以从 $dp[i-1][p][p]$ 转移过来。

以上三种都是会使得答案增加的，当然我们必须得把其余剩下的所有换数情况考虑进来，比如直接把这三个数扔掉；扔掉其中两个数与自己手中的一个数；还有把自己手中的数全扔了，选两个加进来。

这看似是 $O(n^3)$ 的，但是实则每次从 $dp[i-1][][[]]$ 转移到 $dp[i][][[]]$ 的最多是 $O(n)$ 的，所以时间上得到了保证，至于空间上的优化，我们可以直接把第一维压掉，直接覆盖即可（有些当然不能直接覆盖，分析会发现，需要进行转移的信息只有 $O(n)$ 个，所以转移前也需要维护这些信息。）（当然写起来有亿点恶心）

**comment:**当时想到了 $O(n^3)$ 做法，觉得绝对没前途于是放弃，实际上把所有转移方程写出来是可以优化到 $O(n^2)$ 的

**wxg:**

题目大意

**tag:**

做法:

**comment:**

**hxm:**

题目大意：

**tag:**

做法:

comment:

---

## 个人训练

### 傅云濠

比赛□cfglobal#10(ABCDE),abc176(ABCDE)口糊了abc176的F  
补了训练赛的K题,回忆并总结了数位DP□  
整理了几种特殊网络流的板子

---

### 王兴罡

复习了回文自动机,整理字符串模板

---

### 黄旭民

复习了点分治,整理了点分治模板

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

Permanent link:  
[https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:die\\_java:weeksummary12&rev=1598604989](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:die_java:weeksummary12&rev=1598604989)

Last update: 2020/08/28 16:56

