编号	题目	知识点	提交通过率
1	Single	贪心	158/517 (30.56%)
2	powmod	分治	151/619 (24.39%)
3	EatCandy	贪心	124/400 (31.00%)
4	CatAndMouse	贪心/二分	155/410 (37.80%)
5	BinaryTree	搜索回溯/分治	111/277 (40.07%)
6	CandyExchange	搜索回溯/并查集	78/283 (27.56%)
7	ASimpleSequenceProblem	动态规划	39/167 (23.35%)
8	A+B problem	动态规划	5/9 (55.56%)
9	Beautiful	动态规划	0/2 (0.00%)
10	Unblocked project	最小生成树，二分	5/77 (6.49%)
11	Pick	凸包，gcd	0/18 (0.00%)
12	ThinkingProblem	贪心	5/21 (23.81%)
13	DistributedSystem	动态规划，Floyd	2/6 (33.33%)

1.Single

- 可以用一个桶来存储
- 或者根据异或的性质： x xor $x=0$ ，所以把所有数做一次异或就可以了。
- 代码： Single

2.powmod

- 分治
- 代码： powmod

3.EatCandy

- 只能有一次向下走的机会，枚举Alice在第 j 列向下走，那么Bob只能选择 $\sum_{k=j+1}^{m} a_{1,k}$ 或者 $\sum_{k=1}^{j-1} a_{2,k}$ ，Bob选择两者较大的，Alice选择 n 个中较小的。
- 代码： EatCandy

4.CatAndMouse

- 贪心选择优先逃离右边的老鼠即可，模拟一遍
- 代码：CatAndMouse

5.BinaryTree

- 递归分治即可，每次选出最大值，然后再两边分治
- 代码：BinaryTree

6.CandyExchange

- 搜索回溯每个环的大小即可，环的所有答案都是一样的
- 代码：CandyExchange

7.ASimpleSequenceProblem

- 选择翻转奇数长度是无效的
- 翻转偶数长度会造成奇数位置的数到偶数位置，偶数位置的数到奇数位置
- 和相邻的数捆绑，做两次最大子段和即可
- 代码：ASimpleSequenceProblem

8.A+B problem

- 把 n 按照奇偶位置分成两个数 x, y 计算即可，得到 x 有 $0+x, 1+(x-1), \ldots, (x-1)+1, x+1$ 共 $x+1$ 种方法，那么答案就是 $(x+1)(y+1)-2$ ，因为对应总和的第一个数字或第二个数字为 0 。
- 代码：做法1
- 另外还有一种动态规划做法：做法2

9.Beautiful

- 可以发现这是一个树形结构
- a_i 要么取 l_i 要么取 r_i ，取中间值不会更优
- 动态规划即可
- 代码：Beautiful

10.Unblocked project

- 最小（最大）生成树问题
- $\frac{\sum len}{\sum cost}$ 的最大值，考虑二分，$w_i \in \{0, 1\}$ ，表示选或者不选

- $$\begin{aligned} & \frac{\sum len_i \cdot w_i}{\sum cost_i \cdot w_i} \geq mid \\ \Longrightarrow \quad & \sum len_i \cdot w_i \geq mid \cdot \sum cost \cdot w_i \\ \Longrightarrow \quad & w_i(\sum len_i - mid \cdot \sum cost) \geq 0 \end{aligned}$$

- 把 $len - mid \cdot cost$ 作为新的边权，然后做二分判断即可。
- 代码：Unblocked project

11.Pick

- 有了Pick定理我们可以求出两个，用两个算出剩下一个
- 内部格点不好计算，我们使用Pick计算
- 首先我们需要求出这些点的凸包
- 任意一个多边形的面积等于按顺序求相邻两个点与原点组成的向量的叉积之和
- 对于一条边的格点数为 $gcd(dx, dy) + 1$ ，由于是一个闭合图形，所以总的格点数为 $\sum gcd(dx, dy)$ ， dx 或 dy 为 0 时也成立，因为 $gcd(0, x) = x$ 。
- 代码：Pick

12.ThinkingProblem

- 每次加入两个数，那么中位数的变化一定不能移动两次
- 也就是 a_i 和 a_{i-1} 之间不能有其他数，即不存在 $min < a_j < max(1 \leq j \leq i-2)$ ，$min(max)$ 表示 a_i, a_{i-1} 的较大（小）数
- 用set来维护即可
- 不知道为什么有的同学会想到用BIT这样复杂的数据结构来维护（
- 代码：ThinkingProblem

13.DistributedSystem

- 考虑倒着思考，每次加一个点，然后做松弛操作即可，是一个动态规划的过程，类似Floyd
- 代码：DistributedSystem