

## 0801 Solution

---

欢乐乱搞场

### A

直接暴力 next\_permutation 或者暴力搜索即可, 需要模拟一个简单的表达式解析。

时间复杂度  $O(n!)$

### B

考虑直接暴力搜索每个位置放了或者没有放, 可能的优化:

- 如果某一行/某一列超过限制了或者全选了也不够, 直接剪枝
- 从最后一行/最后一列开始搜索, 因为权值大能够对后面有比较大的影响。

时间复杂度  $O(\text{能过})$

### C

我们本质上就是需要构造一个有向图, 我们可以把重边的重数看成边权, 同时上面的字母是不重要的。考虑怎么通过搜索构造这个有向图。

我们需要确定的东西: 终点集合, 边权。

整体搜索的想法就是一个一个点的加上, 搜索连边情况剪掉不用的状态

- 如果某个时刻能够接受的状态已经超过了, 剪枝
- 限制每条边的边权上界
- 在估计某个时刻能够接受的状态的时候加上不能走到终点的点的贡献, 因为这些点一定在之后的某个时刻能够走到终点。
- ...

最后通过限制我们发现所有点都在  $n \leq 4$  有解, 在限制了边权上界为 4 能够很快的求解出答案。

时间复杂度  $O(\text{能过})$

### D

考虑搜索平方之前的数, 我们可以发现如果确定了一个后缀那么平方之后的一个后缀也确定了, 同时如果确定了一个前缀, 平方之后的前缀也能规范到一个区间里面。

于是我们就可以按照后面一个数前面一个数的顺序进行搜索，如果某个时刻确定的后缀没有办法匹配当前的前缀区间就剪枝。

这种做法可以在一个小时之内算出所有平方前  $10^{17}$  的答案，也就是数据范围里的  $k \leq 485$ .