

Eli:

算法 1:

暴力。怎么写我根本都没想，因为这个模型大家都写过不下 3 遍了。

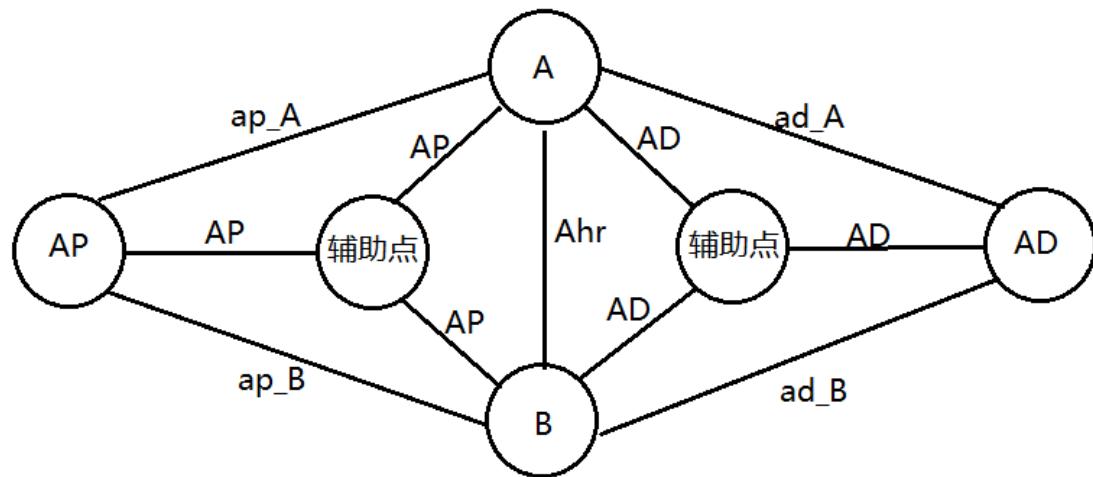
算法 2:

最小割！没有卡时，没有卡内存，更没有邪恶的卡精度卡常数。

注意输入边时是先 AD 后 AP。

做法不唯一，我的方法仅供参考。

对于下述建图大家可以考虑两点同 AP、同 AD、不同时需要的割边。



Caitlyn:

算法 1: [1, 2]

暴搜，搜哪个点是那 S 个点里面的，然后乱搞。

20 分。

算法 2: [6]

没有[和平使者]，那么两种情况，若全偶，答案 1，否则 0。

10 分。

算法 3: [7]

我没有找到这点的特殊性质，但是或许你们没有正解却想出了方法。

我是有多良心!!

10 分。

算法 4: [1, 3, 9]

手玩出那个提供给你的点，然后输出 0。

30 分。

算法 5: [3, 4, 5]

我们发现如果确定有多少行是奇数次[和平使者]，多少列是奇数次[和平使者]，那么奇点个数就确定了！

所以设有 x 个奇行， y 个奇列，然后 $S = x(m - y) + (n - x)y$ ，枚举 x ，则 $y = (S - xm) / (n - 2x)$ ，这样可以算出 y ，注意定义域。

然后对于每对可行的 (x, y) ， x 有 $C(n, x)$ 种放法， y 有 $C(m, y)$ 种放法，剩下的 R 和 C 则需要是偶数，这样可以两两放一行，保证原来的行列的技能次数奇偶性不变。

假设剩下的是 $2r$ 和 $2c$ ，那么根据插板法，答案是

$C(r+n-1, n-1) * C(c+m-1, m-1)$ 。

然后组合数的时间复杂度是上界为 $O(n)$ 。

30 分？或许更多？

算法 6: [8, 9, 10]

我们可以预处理出 [阶乘%mod、逆元、 $1 \sim i$ 的逆元的乘积]

然后组合数就是 $O(1)$ 了，逆元无论线性筛还是 $n \log n$ 欧拉函数都是能过的。

30 分。

Katarina:

算法 1: [1]

注意 m 可能 $=0$

算法 2: [2]

$O(n*n)$ DP

算法 3: [3, 4]

我们可以取所有数中最大的, 然后用一些乱七八糟的线段树啊什么的维护哪些数能取等等。 两个 \log

算法 4: [5, 6]

全取! 然后可能 $Ans=-1$

算法 5: [7]

输出 “-1”!

算法 6: [8, 9, 10]

首先我们可以处理出到每个点时最多能取几个。

然后我们从头到尾扫一遍, 然后把能取的都入优先队列, 一旦超过到该点的限制, 就让最小的出队。

最后剩下的就是能取的所有点中最优集合。