

# Codeforces 662C - Binary Table

[author]

[institute]

February 14, 2019

# 题目描述 & 数据范围

# 题目描述 & 数据范围

现在有一个  $n$  行  $m$  列的表，表中每个格子包含 0 或 1。一次操作可以选择一行或一列并反转所有值。

# 题目描述 & 数据范围

现在有一个  $n$  行  $m$  列的表，表中每个格子包含 0 或 1。一次操作可以选择一行或一列并反转所有值。  
最少能使表中有多少个 1？

# 题目描述 & 数据范围

现在有一个  $n$  行  $m$  列的表，表中每个格子包含 0 或 1。一次操作可以选择一行或一列并反转所有值。  
最少能使表中有多少个 1？

- $1 \leq n \leq 20$

# 题目描述 & 数据范围

现在有一个  $n$  行  $m$  列的表，表中每个格子包含 0 或 1。一次操作可以选择一行或一列并反转所有值。

最少能使表中有多少个 1？

- $1 \leq n \leq 20$
- $1 \leq m \leq 100000$


h?

ん?

独立完成此题即可获得一项神秘成就，还有可能省选翻盘（确信）


ん?

独立完成此题即可获得一项神秘成就，还有可能省选翻盘（确信）

#	Who	=	*	<a href="#">A</a> 500	<a href="#">B</a> 250	<a href="#">C</a> 2250	<a href="#">D</a> 250	<a href="#">E</a> 2250
1	 tourist	2420	+1	464 00:18	217 00:33		145 01:45	1494 01:24

ん?

独立完成此题即可获得一项神秘成就，还有可能省选翻盘（确信）

#	Who	=	*	A 500	B 250	C 2250	D 250	E 2250
1	 tourist	2420	+1	464 00:18	217 00:33		145 01:45	1494 01:24

	NOIP	JLOI day1	JLOI day2	JLOI	加权总分
GXZlegend 没有 CE	520	295	259	554	540.4
Lijinnn	470	210	243	453	459.8
GXZlegend 做出了 D2T3	520	100	295	395	445
Starria	510	150	247	397	442.2
EdwardFrog	490	185	223	408	440.8
CQzhangyu	485	120	283	403	435.8
GXZlegend	520	100	259	359	423.4

JLOI2018 前5。NOIP2017 JL 参赛人数只有99，注意到 €€€ 相关规定，JL 每个学校只有  $\text{round}(5 + \text{round}(99 * 2.5\%)) = 2$  个省队名额。如果 GXZlegend 做出了 D2T3，就能在 day1 CE 两题，day2 水题原题的情况下奇迹翻盘，真实可惜。

# 暴力

# 暴力

枚举每一行是否反转

# 暴力

枚举每一行是否反转  
每一列是独立的，直接贪心决定是否反转

# 暴力

枚举每一行是否反转

每一列是独立的，直接贪心决定是否反转

$O(2^n nm)$

# 优化

# 优化

显然一列可以压成一个二进制数

# 优化

显然一列可以压成一个二进制数  
行的反转相当于所有二进制数按位异或一个数

# 优化

显然一列可以压成一个二进制数  
 行的反转相当于所有二进制数按位异或一个数  
 令第  $j$  列压成的数是  $a_j$ ，那么就是要求

$$\max_{x=0}^{2^n-1} \sum_{j=1}^m \max(\text{popcount}(a_j \oplus x), n - \text{popcount}(a_j \oplus x))$$

# 优化

# 优化

令  $b_i = \sum_{j=1}^m [a_j = i]$ ,  $f_x = \max(\text{popcount}(x), n - \text{popcount}(x))$

# 优化

令  $b_i = \sum_{j=1}^m [a_j = i]$ ,  $f_x = \max(\text{popcount}(x), n - \text{popcount}(x))$

那么现在要求

$$\max_{x=0}^{2^n-1} \sum_{i=0}^{2^n-1} b_i f_{i \oplus x}$$

# 优化

令  $b_i = \sum_{j=1}^m [a_j = i]$ ,  $f_x = \max(\text{popcount}(x), n - \text{popcount}(x))$

那么现在要求

$$\max_{x=0}^{2^n-1} \sum_{i=0}^{2^n-1} b_i f_{i \oplus x}$$

即

$$\max_{x=0}^{2^n-1} \sum_{i=0}^{2^n-1} \sum_{j=0}^{2^n-1} [i \oplus x = j] b_i f_j$$

# 优化

令  $b_i = \sum_{j=1}^m [a_j = i]$ ,  $f_x = \max(\text{popcount}(x), n - \text{popcount}(x))$

那么现在要求

$$\max_{x=0}^{2^n-1} \sum_{i=0}^{2^n-1} b_i f_{i \oplus x}$$

即

$$\begin{aligned} & \max_{x=0}^{2^n-1} \sum_{i=0}^{2^n-1} \sum_{j=0}^{2^n-1} [i \oplus x = j] b_i f_j \\ &= \max_{x=0}^{2^n-1} \sum_{i=0}^{2^n-1} \sum_{j=0}^{2^n-1} [i \oplus j = x] b_i f_j \end{aligned}$$

# 优化

令  $b_i = \sum_{j=1}^m [a_j = i]$ ,  $f_x = \max(\text{popcount}(x), n - \text{popcount}(x))$

那么现在要求

$$\max_{x=0}^{2^n-1} \sum_{i=0}^{2^n-1} b_i f_{i \oplus x}$$

即

$$\begin{aligned} & \max_{x=0}^{2^n-1} \sum_{i=0}^{2^n-1} \sum_{j=0}^{2^n-1} [i \oplus x = j] b_i f_j \\ &= \max_{x=0}^{2^n-1} \sum_{i=0}^{2^n-1} \sum_{j=0}^{2^n-1} [i \oplus j = x] b_i f_j \end{aligned}$$

显然只要求一次 xor 卷积即可对所有  $x$  求出式子后半部分

# 优化

令  $b_i = \sum_{j=1}^m [a_j = i]$ ,  $f_x = \max(\text{popcount}(x), n - \text{popcount}(x))$

那么现在要求

$$\max_{x=0}^{2^n-1} \sum_{i=0}^{2^n-1} b_i f_{i \oplus x}$$

即

$$\begin{aligned} & \max_{x=0}^{2^n-1} \sum_{i=0}^{2^n-1} \sum_{j=0}^{2^n-1} [i \oplus x = j] b_i f_j \\ &= \max_{x=0}^{2^n-1} \sum_{i=0}^{2^n-1} \sum_{j=0}^{2^n-1} [i \oplus j = x] b_i f_j \end{aligned}$$

显然只要求一次 xor 卷积即可对所有  $x$  求出式子后半部分

$O(2^n n)$

# Thanks!