Open Cup选讲

Roundgod

2018 年 11 月 4 日

Unlucky Island

Unlucky Island

- 在一个与世隔绝的小岛上居住着N个人，岛上的人两两之间都有着以下三种关系中的一种：朋友，敌人，或者互不认识。

Unlucky Island

- 在一个与世隔绝的小岛上居住着N个人，岛上的人两两之间都有着以下三种关系中的一种：朋友，敌人，或者互不认识。
- 当岛上的两个人不认识的人A和B相遇时，他们确定关系的过程如下：其中较高的一个人(假设是A)，以某种顺序报出他所认识的人的编号，直到遇到第一个A和B都认识的人C, 然后根据A和C的关系以及B和C的关系来确定A和B的关系：如果A, B和C的关系相同。那么A和B成为朋友，否则A和B成为敌人。

Unlucky Island

- 在一个与世隔绝的小岛上居住着N个人，岛上的人两两之间都有着以下三种关系中的一种：朋友，敌人，或者互不认识。
- 当岛上的两个人不认识的人A和B相遇时，他们确定关系的过程如下：其中较高的一个人(假设是A)，以某种顺序报出他所认识的人的编号，直到遇到第一个A和B都认识的人C, 然后根据A和C的关系以及B和C的关系来确定A和B的关系：如果A, B和C的关系相同。那么A和B成为朋友，否则A和B成为敌人。
- Uef是这个岛上最高的人且编号为1，他现在想要随机找一个其他人，确定他们之间的关系，他希望你能帮他安排一下报编号的顺序，使得其他任意一个人能够与他成为朋友的概率最大。输出这个概率。

Unlucky Island

- 在一个与世隔绝的小岛上居住着N个人，岛上的人两两之间都有着以下三种关系中的一种：朋友，敌人，或者互不认识。
- 当岛上的两个人不认识的人A和B相遇时，他们确定关系的过程如下：其中较高的一个人(假设是A)，以某种顺序报出他所认识的人的编号，直到遇到第一个A和B都认识的人C, 然后根据A和C的关系以及B和C的关系来确定A和B的关系：如果A, B和C的关系相同。那么A和B成为朋友，否则A和B成为敌人。
- Uef是这个岛上最高的人且编号为1，他现在想要随机找一个其他人，确定他们之间的关系，他希望你能帮他安排一下报编号的顺序，使得其他任意一个人能够与他成为朋友的概率最大。输出这个概率。
- $2 \leq N \leq 42, 1 \leq M \leq \frac{N(N-1)}{2}$

Unlucky Island

- 在一个与世隔绝的小岛上居住着N个人，岛上的人两两之间都有着以下三种关系中的一种：朋友，敌人，或者互不认识。
- 当岛上的两个人不认识的人A和B相遇时，他们确定关系的过程如下：其中较高的一个人(假设是A)，以某种顺序报出他所认识的人的编号，直到遇到第一个A和B都认识的人C, 然后根据A和C的关系以及B和C的关系来确定A和B的关系：如果A, B和C的关系相同。那么A和B成为朋友，否则A和B成为敌人。
- Uef是这个岛上最高的人且编号为1，他现在想要随机找一个其他人，确定他们之间的关系，他希望你能帮他安排一下报编号的顺序，使得其他任意一个人能够与他成为朋友的概率最大。输出这个概率。
- $2 \leq N \leq 42, 1 \leq M \leq \frac{N(N-1)}{2}$
- Time Limit: 1.5s Memory Limit: 256MB

Unlucky Island

- 在一个与世隔绝的小岛上居住着N个人，岛上的人两两之间都有着以下三种关系中的一种：朋友，敌人，或者互不认识。
- 当岛上的两个人不认识的人A和B相遇时，他们确定关系的过程如下：其中较高的一个人(假设是A)，以某种顺序报出他所认识的人的编号，直到遇到第一个A和B都认识的人C, 然后根据A和C的关系以及B和C的关系来确定A和B的关系：如果A, B和C的关系相同。那么A和B成为朋友，否则A和B成为敌人。
- Uef是这个岛上最高的人且编号为1，他现在想要随机找一个其他人，确定他们之间的关系，他希望你能帮他安排一下报编号的顺序，使得其他任意一个人能够与他成为朋友的概率最大。输出这个概率。
- $2 \le N \le 42, 1 \le M \le \frac{N(N-1)}{2}$
- Time Limit: 1.5s Memory Limit: 256MB
- Source: XVIII Open Cup named after E.V. Pankratiev. Grand Prix of Eurasia

Unlucky Island

Unlucky Island

- 可以发现$N \leq 42$这个条件有♂点东西

Unlucky Island

- 可以发现$N \leq 42$这个条件有♂点东西
- 除掉Uef之外，我们要考虑的至多只有41个人。如果我们把人分成Uef认识的和不认识的两类，那么其中肯定有某类人的人数≤ 20。

Unlucky Island

- 可以发现$N \leq 42$这个条件有♂点东西
- 除掉Uef之外，我们要考虑的至多只有41个人。如果我们把人分成Uef认识的和不认识的两类，那么其中肯定有某类人的人数≤ 20。
- 第一种情况:Uef认识的人≤ 20

Unlucky Island

- 可以发现$N \leq 42$这个条件有♂点东西
- 除掉Uef之外，我们要考虑的至多只有41个人。如果我们把人分成Uef认识的和不认识的两类，那么其中肯定有某类人的人数≤ 20。
- 第一种情况:Uef认识的人≤ 20
- 使用状压dp，令dp_{mask}表示在已经列出的认识的人的集合为$mask$的情况下，能和Uef成为朋友的最大人数。

Unlucky Island

- 可以发现$N \leq 42$这个条件有♂点东西
- 除掉Uef之外，我们要考虑的至多只有41个人。如果我们把人分成Uef认识的和不认识的两类，那么其中肯定有某类人的人数≤ 20。
- 第一种情况:Uef认识的人≤ 20
- 使用状压dp，令dp_{mask}表示在已经列出的认识的人的集合为$mask$的情况下，能和Uef成为朋友的最大人数。
- 这样转移时只需要枚举下一个列出的人x即可，即$dp_{mask|\{x\}} = \max(dp_{mask} + \#$能通过$x$与Uef成为朋友的人$)$

Unlucky Island

- 可以发现$N \le 42$这个条件有♂点东西
- 除掉Uef之外，我们要考虑的至多只有41个人。如果我们把人分成Uef认识的和不认识的两类，那么其中肯定有某类人的人数≤ 20。
- 第一种情况:Uef认识的人≤ 20
- 使用状压dp，令dp_{mask}表示在已经列出的认识的人的集合为$mask$的情况下，能和Uef成为朋友的最大人数。
- 这样转移时只需要枚举下一个列出的人x即可，即$dp_{mask|\{x\}} = \max(dp_{mask} + \#$能通过$x$与Uef成为朋友的人$)$
- 预处理dp转移式中的后一项，用一个$longlong$型数来存储集合，时间复杂度是$O(20 \cdot 2^{20})$，使用$bitset$的话时间复杂度是$O(\frac{20 \cdot 2^{20}}{wordsize})$

Unlucky Island

Unlucky Island

- 第二种情况:Uef不认识的人≤ 20

Unlucky Island

- 第二种情况:Uef不认识的人≤ 20
- 同样使用状压dp，令dp_{mask}表示和Uef还没有确定关系的人的集合为$mask$的情况下，还能和Uef成为朋友的最大人数。

Unlucky Island

- 第二种情况:Uef不认识的人≤ 20
- 同样使用状压dp，令dp_{mask}表示和Uef还没有确定关系的人的集合为$mask$的情况下，还能和Uef成为朋友的最大人数。
- 之所以可以这样设置dp，是因为存在一个特点：已经被列出的人的集合可以通过还没有确定关系的人的集合唯一确定。

Unlucky Island

- 第二种情况:Uef不认识的人≤ 20
- 同样使用状压dp，令dp_{mask}表示和Uef还没有确定关系的人的集合为$mask$的情况下，还能和Uef成为朋友的最大人数。
- 之所以可以这样设置dp，是因为存在一个特点：已经被列出的人的集合可以通过还没有确定关系的人的集合唯一确定。
- 这样转移时同样只需要枚举下一个列出的人x即可，即$dp_{mask|\{\text{认识}x\text{的集合}\}} =$ $\max(dp_{mask} + \#\text{认识}x\text{的集合中能通过}x\text{与Uef成为朋友的人})$

Unlucky Island

- 第二种情况:Uef不认识的人≤ 20
- 同样使用状压dp，令dp_{mask}表示和Uef还没有确定关系的人的集合为$mask$的情况下，还能和Uef成为朋友的最大人数。
- 之所以可以这样设置dp，是因为存在一个特点：已经被列出的人的集合可以通过还没有确定关系的人的集合唯一确定。
- 这样转移时同样只需要枚举下一个列出的人x即可，
  即$dp_{mask|\{认识x的集合\}} =$
  $\max(dp_{mask} + \#认识x的集合中能通过x与\text{Uef}成为朋友的人)$
- 预处理dp转移式中的后一项，用一个$longlong$型数来存储集合，时间复杂度是$O(20 \cdot 2^{20})$，使用$bitset$的话时间复杂度是$O(\frac{20 \cdot 2^{20}}{wordsize})$

Anna and Lucky Tickets

- Anna有一个长度为n的数字串，她想要知道这串数字是否是幸运的。她有两种验证的方法：

Anna and Lucky Tickets

- Anna有一个长度为n的数字串，她想要知道这串数字是否是幸运的。她有两种验证的方法：
- 如果这个数字串奇数位置上的数字和等于偶数位置上的数字和，那么Anna认为这个数字串是幸运的。

Anna and Lucky Tickets

- Anna有一个长度为n的数字串，她想要知道这串数字是否是幸运的。她有两种验证的方法：
- 如果这个数字串奇数位置上的数字和等于偶数位置上的数字和，那么Anna认为这个数字串是幸运的。
- 如果这个数字串前$\lfloor \frac{n}{2} \rfloor$位置上的数字和等于后$\lfloor \frac{n}{2} \rfloor$位置上的数字和，那么Anna也认为这个数字串是幸运的。

Anna and Lucky Tickets

- Anna有一个长度为n的数字串，她想要知道这串数字是否是幸运的。她有两种验证的方法：
- 如果这个数字串奇数位置上的数字和等于偶数位置上的数字和，那么Anna认为这个数字串是幸运的。
- 如果这个数字串前$\lfloor \frac{n}{2} \rfloor$位置上的数字和等于后$\lfloor \frac{n}{2} \rfloor$位置上的数字和，那么Anna也认为这个数字串是幸运的。
- 然而，如果两种都满足，那么Anna会非常害怕以至于她会认为这个数字串是不幸运的。如果两种都不满足，那么Anna也会认为这个数字串是不幸运的。现在Anna想知道，有多少个**回文**数字串是不幸运的。

Anna and Lucky Tickets

- Anna有一个长度为n的数字串，她想要知道这串数字是否是幸运的。她有两种验证的方法：
- 如果这个数字串奇数位置上的数字和等于偶数位置上的数字和，那么Anna认为这个数字串是幸运的。
- 如果这个数字串前$\lfloor \frac{n}{2} \rfloor$位置上的数字和等于后$\lfloor \frac{n}{2} \rfloor$位置上的数字和，那么Anna也认为这个数字串是幸运的。
- 然而，如果两种都满足，那么Anna会非常害怕以至于她会认为这个数字串是不幸运的。如果两种都不满足，那么Anna也会认为这个数字串是不幸运的。现在Anna想知道，有多少个**回文**数字串是不幸运的。
- $2 \leq N \leq 10^6$

Anna and Lucky Tickets

- Anna有一个长度为n的数字串，她想要知道这串数字是否是幸运的。她有两种验证的方法：
- 如果这个数字串奇数位置上的数字和等于偶数位置上的数字和，那么Anna认为这个数字串是幸运的。
- 如果这个数字串前$\lfloor \frac{n}{2} \rfloor$位置上的数字和等于后$\lfloor \frac{n}{2} \rfloor$位置上的数字和，那么Anna也认为这个数字串是幸运的。
- 然而，如果两种都满足，那么Anna会非常害怕以至于她会认为这个数字串是不幸运的。如果两种都不满足，那么Anna也会认为这个数字串是不幸运的。现在Anna想知道，有多少个**回文**数字串是不幸运的。
- $2 \leq N \leq 10^6$
- Time Limit: 2s Memory Limit: 256MB

Anna and Lucky Tickets

- Anna有一个长度为n的数字串，她想要知道这串数字是否是幸运的。她有两种验证的方法：
- 如果这个数字串奇数位置上的数字和等于偶数位置上的数字和，那么Anna认为这个数字串是幸运的。
- 如果这个数字串前$\lfloor \frac{n}{2} \rfloor$位置上的数字和等于后$\lfloor \frac{n}{2} \rfloor$位置上的数字和，那么Anna也认为这个数字串是幸运的。
- 然而，如果两种都满足，那么Anna会非常害怕以至于她会认为这个数字串是不幸运的。如果两种都不满足，那么Anna也会认为这个数字串是不幸运的。现在Anna想知道，有多少个**回文**数字串是不幸运的。
- $2 \leq N \leq 10^6$
- Time Limit: 2s Memory Limit: 256MB
- Source: XVIII Open Cup named after E.V. Pankratiev. Grand Prix of Urals

Anna and Lucky Tickets

- 很容易发现，对于一个回文串，那么其前$\lfloor \frac{n}{2} \rfloor$位置上的数字和一定等于后$\lfloor \frac{n}{2} \rfloor$位置上的数字和。

Anna and Lucky Tickets

- 很容易发现，对于一个回文串，那么其前$\lfloor \frac{n}{2} \rfloor$位置上的数字和一定等于后$\lfloor \frac{n}{2} \rfloor$位置上的数字和。
- 那么我们就是要求出奇数位置上的数字和等于偶数位置上的数字和的回文串。

Anna and Lucky Tickets

- 很容易发现，对于一个回文串，那么其前$\lfloor \frac{n}{2} \rfloor$位置上的数字和一定等于后$\lfloor \frac{n}{2} \rfloor$位置上的数字和。
- 那么我们就是要求出奇数位置上的数字和等于偶数位置上的数字和的回文串。
- 如果N是偶数，那么答案是什么？

Anna and Lucky Tickets

- 很容易发现，对于一个回文串，那么其前$\lfloor \frac{n}{2} \rfloor$位置上的数字和一定等于后$\lfloor \frac{n}{2} \rfloor$位置上的数字和。
- 那么我们就是要求出奇数位置上的数字和等于偶数位置上的数字和的回文串。
- 如果N是偶数，那么答案是什么？
- $10^{N/2}$

Anna and Lucky Tickets

- 很容易发现，对于一个回文串，那么其前$\lfloor \frac{n}{2} \rfloor$位置上的数字和一定等于后$\lfloor \frac{n}{2} \rfloor$位置上的数字和。
- 那么我们就是要求出奇数位置上的数字和等于偶数位置上的数字和的回文串。
- 如果N是偶数，那么答案是什么？
- $10^{N/2}$

Anna and Lucky Tickets

- 问题在于N是奇数的情况。

Anna and Lucky Tickets

- 问题在于N是奇数的情况。
- 为了方便起见，我们假设位于中间的数在奇数位置上，在偶数位置上的解法类似。

Anna and Lucky Tickets

- 问题在于N是奇数的情况。
- 为了方便起见，我们假设位于中间的数在奇数位置上，在偶数位置上的解法类似。
- 相当于要求出$0 \leq 2x_1 - 2x_2 + 2x_3 - ...2x_{\lfloor \frac{N}{2} \rfloor} \leq 9$的方案数，其中$0 \leq x_i \leq 9$

Anna and Lucky Tickets

- 问题在于N是奇数的情况。
- 为了方便起见，我们假设位于中间的数在奇数位置上，在偶数位置上的解法类似。
- 相当于要求出$0 \le 2x_1 - 2x_2 + 2x_3 - ...2x_{\lfloor \frac{N}{2} \rfloor} \le 9$的方案数，其中$0 \le x_i \le 9$
- 也就是$0 \le x_1 - x_2 + x_3 - ...x_{\lfloor \frac{N}{2} \rfloor} \le 4$的方案数

Anna and Lucky Tickets

- 问题在于N是奇数的情况。
- 为了方便起见，我们假设位于中间的数在奇数位置上，在偶数位置上的解法类似。
- 相当于要求出$0 \leq 2x_1 - 2x_2 + 2x_3 - ...2x_{\lfloor \frac{N}{2} \rfloor} \leq 9$的方案数，其中$0 \leq x_i \leq 9$
- 也就是$0 \leq x_1 - x_2 + x_3 - ...x_{\lfloor \frac{N}{2} \rfloor} \leq 4$的方案数

Anna and Lucky Tickets

Anna and Lucky Tickets

- 2018 Multi-University Training Contest 8, Problem 1, Character Encoding:求方程$x_1 + x_2 + ... + x_n = k$的解数，其中$0 \le x_i \le m$. 数据保证$\sum n \le 5 \times 10^6, \sum m \le 5 \times 10^6, \sum k \le 5 \times 10^6$.

Anna and Lucky Tickets

- 2018 Multi-University Training Contest 8, Problem 1, Character Encoding:求方程$x_1 + x_2 + ... + x_n = k$的解数，其中$0 \leq x_i \leq m$. 数据保证$\sum n \leq 5 \times 10^6, \sum m \leq 5 \times 10^6, \sum k \leq 5 \times 10^6$.
- 这题怎么做?

Anna and Lucky Tickets

- 2018 Multi-University Training Contest 8, Problem 1, Character Encoding:求方程$x_1+x_2+...+x_n=k$的解数，其中$0 \le x_i \le m$. 数据保证$\sum n \le 5 \times 10^6, \sum m \le 5 \times 10^6, \sum k \le 5 \times 10^6$.
- 这题怎么做?
- 首先要知道$x_1+x_2+...+x_n=k$的非负整数解的个数为$\binom{n+k-1}{k}$

Anna and Lucky Tickets

- 2018 Multi-University Training Contest 8, Problem 1, Character Encoding:求方程$x_1 + x_2 + ... + x_n = k$的解数，其中$0 \le x_i \le m$. 数据保证$\sum n \le 5 \times 10^6, \sum m \le 5 \times 10^6, \sum k \le 5 \times 10^6$.
- 这题怎么做?
- 首先要知道$x_1 + x_2 + ... + x_n = k$的非负整数解的个数为$\binom{n+k-1}{k}$
- 假设有i个变量超出限制，那么原方程变为$x_1 + x_2 + ... + x_n = k + (m+1)i$，这样的方程解数为$\binom{n+k+(m+1)i-1}{k}$。

Anna and Lucky Tickets

- 2018 Multi-University Training Contest 8, Problem 1, Character Encoding:求方程$x_1+x_2+...+x_n=k$的解数，其中$0\le x_i\le m$. 数据保证$\sum n\le 5\times 10^6, \sum m\le 5\times 10^6, \sum k\le 5\times 10^6$.
- 这题怎么做？
- 首先要知道$x_1+x_2+...+x_n=k$的非负整数解的个数为$\binom{n+k-1}{k}$
- 假设有i个变量超出限制，那么原方程变为$x_1+x_2+...+x_n=k+(m+1)i$，这样的方程解数为$\binom{n+k+(m+1)i-1}{k}$。
- 那么根据容斥原理，原方程的解数为：$\sum\limits_{i=0}^{n}(-1)^i\binom{n}{i}\binom{n+k+(m+1)i-1}{k}$。

Anna and Lucky Tickets

- 2018 Multi-University Training Contest 8, Problem 1, Character Encoding:求方程$x_1+x_2+...+x_n=k$的解数，其中$0 \le x_i \le m$. 数据保证$\sum n \le 5\times 10^6, \sum m \le 5\times 10^6, \sum k \le 5\times 10^6$.
- 这题怎么做?
- 首先要知道$x_1+x_2+...+x_n=k$的非负整数解的个数为$\binom{n+k-1}{k}$
- 假设有i个变量超出限制，那么原方程变为$x_1+x_2+...+x_n=k+(m+1)i$，这样的方程解数为$\binom{n+k+(m+1)i-1}{k}$。
- 那么根据容斥原理，原方程的解数为：$\sum\limits_{i=0}^{n}(-1)^i\binom{n}{i}\binom{n+k+(m+1)i-1}{k}$。
- 预处理阶乘后时间复杂度$O(n)$.

Anna and Lucky Tickets

Anna and Lucky Tickets

- 回到原来的题目，我们要求出$0 \le x_1 - x_2 + x_3 - ...x_{\lfloor \frac{N}{2} \rfloor} \le 4$的方案数

Anna and Lucky Tickets

- 回到原来的题目，我们要求出$0 \le x_1 - x_2 + x_3 - ...x_{\lfloor \frac{N}{2} \rfloor} \le 4$的方案数
- 因为$x_i \in [0,9]$,我们给每个前带上负号的变量加上9，就变成了$0 \le x_1 + x_2 + x_3 + ... + x_{\lfloor \frac{N}{2} \rfloor} \le 4 + 9j$的方案数，$j$是带负号的变量数(也就是偶数位上的数的个数)

Anna and Lucky Tickets

- 回到原来的题目，我们要求出$0 \le x_1 - x_2 + x_3 - ... x_{\lfloor \frac{N}{2} \rfloor} \le 4$的方案数
- 因为$x_i \in [0, 9]$,我们给每个前带上负号的变量加上9，就变成了$0 \le x_1 + x_2 + x_3 + ... + x_{\lfloor \frac{N}{2} \rfloor} \le 4 + 9j$的方案数，$j$是带负号的变量数(也就是偶数位上的数的个数)
- 遍历不等号后面$[0, 4]$中的所有可能性，用之前容斥原理的方法计算方案数。

Anna and Lucky Tickets

- 回到原来的题目，我们要求出$0 \le x_1 - x_2 + x_3 - ...x_{\lfloor \frac{N}{2} \rfloor} \le 4$的方案数
- 因为$x_i \in [0, 9]$,我们给每个前带上负号的变量加上9，就变成了$0 \le x_1 + x_2 + x_3 + ... + x_{\lfloor \frac{N}{2} \rfloor} \le 4 + 9j$的方案数，$j$是带负号的变量数(也就是偶数位上的数的个数)
- 遍历不等号后面$[0, 4]$中的所有可能性，用之前容斥原理的方法计算方案数。
- 时间复杂度$O(N)$.

Blazing New Trails

- 给定一个N个顶点，M条边的无向图，每条边的边权是w_i。其中有K个顶点是特殊的，你现在需要选择一些边的集合，使得这N个顶点两两之间**恰有一条路径**可以相互到达，而且恰好有W条边连接一个特殊顶点与一个非特殊顶点。请你最小化你选择的边的权值和。如果不存在这样的方案，输出-1。

Blazing New Trails

- 给定一个N个顶点，M条边的无向图，每条边的边权是w_i。其中有K个顶点是特殊的，你现在需要选择一些边的集合，使得这N个顶点两两之间**恰有一条路径**可以相互到达，而且恰好有W条边连接一个特殊顶点与一个非特殊顶点。请你最小化你选择的边的权值和。如果不存在这样的方案，输出-1。
- $2 \leq N \leq 2 \times 10^5$,$1 \leq M \leq 5 \times 10^5$,$1 \leq w_i \leq 10^5$,$1 \leq K \leq N-1$, $1 \leq W \leq N-1$

Blazing New Trails

- 给定一个N个顶点，M条边的无向图，每条边的边权是w_i。其中有K个顶点是特殊的，你现在需要选择一些边的集合，使得这N个顶点两两之间**恰有一条路径**可以相互到达，而且恰好有W条边连接一个特殊顶点与一个非特殊顶点。请你最小化你选择的边的权值和。如果不存在这样的方案，输出-1。
- $2 \le N \le 2 \times 10^5$,$1 \le M \le 5 \times 10^5$,$1 \le w_i \le 10^5$,$1 \le K \le N - 1$, $1 \le W \le N - 1$
- Time Limit: 7s Memory Limit: 512MB

Blazing New Trails

- 给定一个N个顶点，M条边的无向图，每条边的边权是w_i。其中有K个顶点是特殊的，你现在需要选择一些边的集合，使得这N个顶点两两之间**恰有一条路径**可以相互到达，而且恰好有W条边连接一个特殊顶点与一个非特殊顶点。请你最小化你选择的边的权值和。如果不存在这样的方案，输出-1。
- $2 \le N \le 2 \times 10^5$,$1 \le M \le 5 \times 10^5$,$1 \le w_i \le 10^5$,$1 \le K \le N-1$, $1 \le W \le N-1$
- Time Limit: 7s Memory Limit: 512MB
- Source: XVII Open Cup named after E.V. Pankratiev. Grand Prix of America (NAIPC-2017)

Blazing New Trails

- 容易发现题目要求就是求出限制恰有W条特殊边情形下的最小生成树。

Blazing New Trails

- 容易发现题目要求就是求出限制恰有W条特殊边情形下的最小生成树。
- 初看起来十分难以入手。

Blazing New Trails

- 容易发现题目要求就是求出限制恰有W条特殊边情形下的最小生成树。
- 初看起来十分难以入手。
- 为了方便，接下来我们把非特殊边称作黑边，把特殊边称作白边。

Blazing New Trails

- 容易发现题目要求就是求出限制恰有W条特殊边情形下的最小生成树。
- 初看起来十分难以入手。
- 为了方便，接下来我们把非特殊边称作黑边，把特殊边称作白边。
- 假设我们直接跑最小生成树算法，得到的结果有可能不满足恰有W条白边的限制。

Blazing New Trails

- 容易发现题目要求就是求出限制恰有W条特殊边情形下的最小生成树。
- 初看起来十分难以入手。
- 为了方便，接下来我们把非特殊边称作黑边，把特殊边称作白边。
- 假设我们直接跑最小生成树算法，得到的结果有可能不满足恰有W条白边的限制。
- 准确的说，如果白边权值较小，我们选的白边数量很可能会多于W。如果白边权值较大，我们选的白边数量很可能会小于W。

Blazing New Trails

- 给每条白边的权值加x，求出最小生成树，二分使得最小生成树中白边数量恰为W条的x。

Blazing New Trails

- 给每条白边的权值加x，求出最小生成树，二分使得最小生成树中白边数量恰为W条的x。
- 为了一致性，在遇到黑白边权值相等时优先取黑边。

Blazing New Trails

- 给每条白边的权值加x，求出最小生成树，二分使得最小生成树中白边数量恰为W条的x。
- 为了一致性，在遇到黑白边权值相等时优先取黑边。
- 计算出结果后减去Wx即是最终的答案。

Blazing New Trails

- 给每条白边的权值加x，求出最小生成树，二分使得最小生成树中白边数量恰为W条的x。
- 为了一致性，在遇到黑白边权值相等时优先取黑边。
- 计算出结果后减去Wx即是最终的答案。
- 时间复杂度$O(M \log M \log C)$，其中C是二分的上下界差的绝对值。

Blazing New Trails

- 给每条白边的权值加x，求出最小生成树，二分使得最小生成树中白边数量恰为W条的x。
- 为了一致性，在遇到黑白边权值相等时优先取黑边。
- 计算出结果后减去Wx即是最终的答案。
- 时间复杂度$O(M \log M \log C)$，其中C是二分的上下界差的绝对值。
- 还可以做一些优化，做出只用白边和只用黑边的最小生成树。先将黑白边分别排好序，每次归并出整体的顺序，这样的时间复杂度是$O(M \log M + M \log C)$.

Blazing New Trails

- 那么问题来了

Blazing New Trails

- 那么问题来了
- 为什么这样做是对的？

Blazing New Trails

- 那么问题来了
- 为什么这样做是对的？

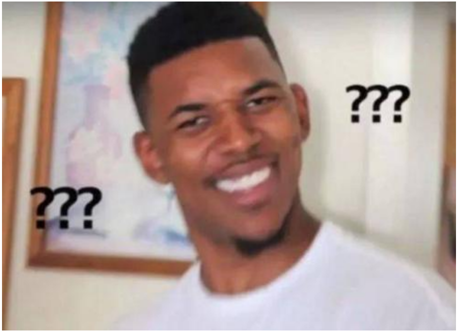

Figure: ???

Blazing New Trails

- 那么问题来了
- 为什么这样做是对的？

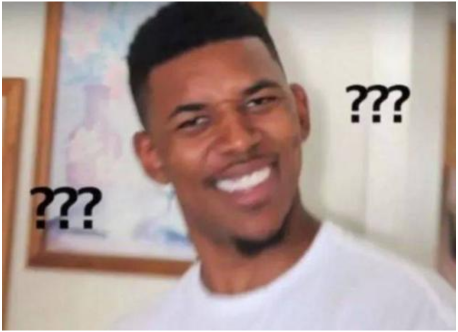

Figure: ???

- 算法竞赛不需要证明(x

Blazing New Trails

- 这实际上是一类方法，把限制个数的最优化问题转化为不限制个数的最优化问题，又被叫做wqs二分，或者Aliens Trick(因为在IOI2016Aliens中出现了这样的方法)

Blazing New Trails

- 这实际上是一类方法，把限制个数的最优化问题转化为不限制个数的最优化问题，又被叫做wqs二分，或者Aliens Trick(因为在IOI2016Aliens中出现了这样的方法)
- 这种方法也并非可以随意使用，它要求答案函数有凸性(convexity)

Blazing New Trails

- 这实际上是一类方法，把限制个数的最优化问题转化为不限制个数的最优化问题，又被叫做wqs二分，或者Aliens Trick(因为在IOI2016Aliens中出现了这样的方法)
- 这种方法也并非可以随意使用，它要求答案函数有凸性(convexity)
- 也就是$ans(i-1) - ans(i) \geq ans(i) - ans(i+1)$

Blazing New Trails

- 这实际上是一类方法，把限制个数的最优化问题转化为不限制个数的最优化问题，又被叫做wqs二分，或者Aliens Trick(因为在IOI2016Aliens中出现了这样的方法)
- 这种方法也并非可以随意使用，它要求答案函数有凸性(convexity)
- 也就是$ans(i-1)-ans(i) \geq ans(i)-ans(i+1)$
- 这看起来就是对的(逃

Blazing New Trails

- 这实际上是一类方法，把限制个数的最优化问题转化为不限制个数的最优化问题，又被叫做wqs二分，或者Aliens Trick(因为在IOI2016Aliens中出现了这样的方法)
- 这种方法也并非可以随意使用，它要求答案函数有凸性(convexity)
- 也就是$ans(i-1) - ans(i) \geq ans(i) - ans(i+1)$
- 这看起来就是对的(逃
- 然而很多题目凸性都不那么容易证明，所以直接莽就对了(x

Xormites

- 给定一个长度为N的序列$A_1, A_2, ...A_N$，两个玩家轮流取走序列最左端的一个数或者序列最右端的一个数，每方的分数是自己取走的所有数的异或(exclusive or,$\oplus$)之和，最终分数高的那一方获胜。请你判断是先手胜，后手胜，或者是平局。

Xormites

- 给定一个长度为N的序列$A_1, A_2, ...A_N$，两个玩家轮流取走序列最左端的一个数或者序列最右端的一个数，每方的分数是自己取走的所有数的异或(exclusive or,$\oplus$)之和，最终分数高的那一方获胜。请你判断是先手胜，后手胜，或者是平局。
- $1 \le N \le 10^5$,$1 \le A_i \le 10^9$

Xormites

- 给定一个长度为N的序列$A_1, A_2, ...A_N$，两个玩家轮流取走序列最左端的一个数或者序列最右端的一个数，每方的分数是自己取走的所有数的异或(exclusive or,$\oplus$)之和，最终分数高的那一方获胜。请你判断是先手胜，后手胜，或者是平局。
- $1 \le N \le 10^5$,$1 \le A_i \le 10^9$
- Time Limit: 1s Memory Limit: 256MB

Xormites

- 给定一个长度为N的序列$A_1, A_2, ...A_N$，两个玩家轮流取走序列最左端的一个数或者序列最右端的一个数，每方的分数是自己取走的所有数的异或(exclusive or,$\oplus$)之和，最终分数高的那一方获胜。请你判断是先手胜，后手胜，或者是平局。
- $1 \leq N \leq 10^5$,$1 \leq A_i \leq 10^9$
- Time Limit: 1s Memory Limit: 256MB
- Source: XVIII Open Cup named after E.V. Pankratiev. Grand Prix of Romania

Xormites

- 首先可以观察到，如果所有数字的异或和是S，第一个人拿的数字的异或和是A，第二个人拿的数字的异或和是B，那么一定有$A \oplus B = S$

Xormites

- 首先可以观察到，如果所有数字的异或和是S，第一个人拿的数字的异或和是A，第二个人拿的数字的异或和是B，那么一定有$A \oplus B = S$
- 因此，平局的情况当且仅当$S = 0$的时候出现

Xormites

- 首先可以观察到，如果所有数字的异或和是S，第一个人拿的数字的异或和是A，第二个人拿的数字的异或和是B，那么一定有$A \oplus B = S$
- 因此，平局的情况当且仅当$S = 0$的时候出现
- 若$S \neq 0$，设i为S的二进制位中为1的最高位，那么最终所取数字异或和第i位为1的人获胜。

Xormites

- 首先可以观察到，如果所有数字的异或和是S，第一个人拿的数字的异或和是A，第二个人拿的数字的异或和是B，那么一定有$A \oplus B = S$
- 因此，平局的情况当且仅当$S = 0$的时候出现
- 若$S \neq 0$，设i为S的二进制位中为1的最高位，那么最终所取数字异或和第i位为1的人获胜。
- 题目转化为以下问题：给定一个长度为N的01序列$A_1, A_2, ...A_N$，其中有奇数个1,两个玩家轮流取走序列最左端的一个数或者序列最右端的一个数，最终取到1的个数为奇数的人获胜。

Xormites

- 如果N为偶数，那么先手必胜。

Xormites

- 如果N为偶数，那么先手必胜。
- 为什么？

Xormites

- 如果N为偶数，那么先手必胜。
- 为什么？
- 将原数列黑白染色，那么先手一定能够取到所有的黑色或者所有的白色。

Xormites

- 如果N为偶数，那么先手必胜。
- 为什么？
- 将原数列黑白染色，那么先手一定能够取到所有的黑色或者所有的白色。
- 所以先手一定能够取到奇数个1.

Xormites

Xormites

- 类似地，如果N为奇数，那么先手第一次必须取1，否则便会成为偶数情况下的后手。

Xormites

- 类似地，如果N为奇数，那么先手第一次必须取1，否则便会成为偶数情况下的后手。
- 接下来，先手每一步必须模仿后手的取法，否则也会转化成之前的情况。

Xormites

- 类似地，如果N为奇数，那么先手第一次必须取1，否则便会成为偶数情况下的后手。
- 接下来，先手每一步必须模仿后手的取法，否则也会转化成之前的情况。
- 在什么情况下先手可以模仿后手的取法呢？

Xormites

- 类似地，如果N为奇数，那么先手第一次必须取1，否则便会成为偶数情况下的后手。
- 接下来，先手每一步必须模仿后手的取法，否则也会转化成之前的情况。
- 在什么情况下先手可以模仿后手的取法呢？
- 如果先手能模仿后手的取法，那么这个串一定是ABA'的形式，其中A'是A的reverse，且在B中数字成对出现(例如001111001100)

Xormites

- 类似地，如果N为奇数，那么先手第一次必须取1，否则便会成为偶数情况下的后手。
- 接下来，先手每一步必须模仿后手的取法，否则也会转化成之前的情况。
- 在什么情况下先手可以模仿后手的取法呢？
- 如果先手能模仿后手的取法，那么这个串一定是ABA¸的形式，其中A'是A的reverse，且在B中数字成对出现(例如001111001100)
- 证明留做习题(逃

Xormites

- 类似地，如果N为奇数，那么先手第一次必须取1，否则便会成为偶数情况下的后手。
- 接下来，先手每一步必须模仿后手的取法，否则也会转化成之前的情况。
- 在什么情况下先手可以模仿后手的取法呢？
- 如果先手能模仿后手的取法，那么这个串一定是ABA'的形式，其中A'是A的reverse，且在B中数字成对出现(例如001111001100)
- 证明留做习题(逃
- 这样，枚举先手第一次取的1是在左端还是右端，分别check一遍，总的时间复杂度$O(n)$.