基于线性代数的一般图匹配

Wearry

May 29, 2017

预备知识

预备知识

记一个 $n \times n$ 的方阵 A 的行列式为 $\det A$, 且:

$$\det A = \sum_{\pi \in \sum_n} (-1)^{sgn\ \pi} \prod_{k=1}^{n} A_{k,\pi(k)}$$

预备知识

记一个 $n \times n$ 的方阵 A 的行列式为 $\det A$, 且:

$$\det A = \sum_{\pi \in \sum_n} (-1)^{sgn\ \pi} \prod_{k=1}^{n} A_{k,\pi(k)}$$

- 其中 π 是 $1-n$ 的一个排列.
- $sgn\ \pi$ 表示 π 的符号, 可以认为是该排列的逆序对个数

预备知识

给定 m 个 n 维向量,$\{v_1, v_2, \cdot, v_m\}$.

预备知识

给定 m 个 n 维向量,$\{v_1, v_2, \cdot, v_m\}$.

- 如果存在 $\{\lambda_1, \lambda_2, \cdot, \lambda_m\}$, 使得 $v = \sum_{i=1}^{m} \lambda_i v_i$, 则称 v 是 v_i 的线性组合.

预备知识

给定 m 个 n 维向量,$\{v_1, v_2, \cdot, v_m\}$.

- 如果存在 $\{\lambda_1, \lambda_2, \cdot, \lambda_m\}$, 使得 $v = \sum_{i=1}^{m} \lambda_i v_i$, 则称 v 是 v_i 的**线性组合**.
- 如果这 m 个向量均不能表示成其它向量的线性组合, 则称这些向量**线性无关**, 否则称之为**线性相关**.

预备知识

给定 m 个 n 维向量,$\{v_1, v_2, \cdot, v_m\}$.

- 如果存在 $\{\lambda_1, \lambda_2, \cdot, \lambda_m\}$, 使得 $v = \sum_{i=1}^{m} \lambda_i v_i$, 则称 v 是 v_i 的**线性组合**.
- 如果这 m 个向量均不能表示成其它向量的线性组合, 则称这些向量**线性无关**, 否则称之为**线性相关**.

矩阵的秩

对于一个矩阵,
将其每一行看做一个向量, 从中选出的极大线性无关向量数称为**行秩**,
将其每一列看做一个向量, 从中选出的极大线性无关向量数称为**列秩**,

预备知识

给定 m 个 n 维向量,$\{v_1, v_2, \cdot, v_m\}$.

- 如果存在 $\{\lambda_1, \lambda_2, \cdot, \lambda_m\}$, 使得 $v = \sum_{i=1}^{m} \lambda_i v_i$, 则称 v 是 v_i 的**线性组合**.
- 如果这 m 个向量均不能表示成其它向量的线性组合, 则称这些向量**线性无关**, 否则称之为**线性相关**.

矩阵的秩

对于一个矩阵,
将其每一行看做一个向量, 从中选出的极大线性无关向量数称为**行秩**,
将其每一列看做一个向量, 从中选出的极大线性无关向量数称为**列秩**,
不难发现行秩与列秩总是相等的, 记为 $rank\ A$.

完美匹配的存在性

完美匹配的存在性

定义一个图 G 的 $Tutte$ 矩阵 $\tilde{A}(G)$, 其中:

$$\tilde{A}(G)_{i,j} = \begin{cases} x_{i,j}, & (i,j) \in E, i < j \\ -x_{i,j}, & (i,j) \in E, i > j \\ 0, & otherwise \end{cases}$$

完美匹配的存在性

定义一个图 G 的 $Tutte$ 矩阵 $\tilde{A}(G)$, 其中:

$$\tilde{A}(G)_{i,j} = \begin{cases} x_{i,j}, & (i,j) \in E, i < j \\ -x_{i,j}, & (i,j) \in E, i > j \\ 0, & otherwise \end{cases}$$

$Tutte$ 定理: 图 G 存在完美匹配当且仅当 $\det \tilde{A}(G) \neq 0$.

完美匹配的存在性

简单地证明一下:

引理

- $\det \tilde{A}(G)$ 中每一项对应图中一个环覆盖。
- 图 G 存在完美匹配当且仅当其有偶环覆盖

完美匹配的存在性

简单地证明一下:

引理

- $\det \tilde{A}(G)$ 中每一项对应图中一个环覆盖。
- 图 G 存在完美匹配当且仅当其有偶环覆盖

对于 $\det \tilde{A}(G)$ 中的每一项.
将 π 用 π' 代替, 其中 π' 是 π 翻转一个环覆盖得到的排列.

完美匹配的存在性

简单地证明一下:

引理

- $\det \tilde{A}(G)$ 中每一项对应图中一个环覆盖。
- 图 G 存在完美匹配当且仅当其有偶环覆盖

对于 $\det \tilde{A}(G)$ 中的每一项.
将 π 用 π' 代替, 其中 π' 是 π 翻转一个环覆盖得到的排列.

不难发现 $sgn(\pi) = sgn(\pi')$.

完美匹配的存在性

简单地证明一下:

引理

- $\det \tilde{A}(G)$ 中每一项对应图中一个环覆盖。
- 图 G 存在完美匹配当且仅当其有偶环覆盖

对于 $\det \tilde{A}(G)$ 中的每一项.
将 π 用 π' 代替, 其中 π' 是 π 翻转一个环覆盖得到的排列.

不难发现 $sgn(\pi) = sgn(\pi')$.
于是如果 π 中存在奇环, 它对 $\det \tilde{A}(G)$ 的贡献就为 0.

完美匹配的存在性

简单地证明一下:

引理

- $\det \tilde{A}(G)$ 中每一项对应图中一个环覆盖。
- 图 G 存在完美匹配当且仅当其有偶环覆盖

对于 $\det \tilde{A}(G)$ 中的每一项.
将 π 用 π' 代替, 其中 π' 是 π 翻转一个环覆盖得到的排列.

不难发现 $sgn(\pi) = sgn(\pi')$.
于是如果 π 中存在奇环, 它对 $\det \tilde{A}(G)$ 的贡献就为 0.

完美匹配的存在性

然而 $Tutte$ 矩阵中的元素均为未知数, 如何判定是否恒为 0 呢？

完美匹配的存在性

然而 $Tutte$ 矩阵中的元素均为未知数, 如何判定是否恒为 0 呢？

我们考虑如下随机算法:

完美匹配的存在性

然而 $Tutte$ 矩阵中的元素均为未知数, 如何判定是否恒为 0 呢？

我们考虑如下随机算法:

取一个 10^9 级别的大质数 p, 每个 $x_{i,j}$ 在 $mod\ p$ 下随机取值. 然后求行列式的值并判定是否存在完美匹配.

完美匹配的存在性

然而 $Tutte$ 矩阵中的元素均为未知数, 如何判定是否恒为 0 呢？

我们考虑如下随机算法:

取一个 10^9 级别的大质数 p, 每个 $x_{i,j}$ 在 $mod\ p$ 下随机取值. 然后求行列式的值并判定是否存在完美匹配.

由代数相关知识, 这样计算出错的概率不到 10^{-6}.

构造完美匹配方案

构造完美匹配方案

首先考虑如下暴力：

- 枚举一条边 $(u, v) \in E$ 并将其从原图中删除.

构造完美匹配方案

首先考虑如下暴力：

- 枚举一条边 $(u, v) \in E$ 并将其从原图中删除.
- 判定删除后的边是否存在完美匹配.

构造完美匹配方案

首先考虑如下暴力：

- 枚举一条边 $(u, v) \in E$ 并将其从原图中删除.
- 判定删除后的边是否存在完美匹配.
- 不断重复这个过程直到完成.

构造完美匹配方案

首先考虑如下暴力:

- 枚举一条边 $(u, v) \in E$ 并将其从原图中删除.
- 判定删除后的边是否存在完美匹配.
- 不断重复这个过程直到完成.

这样做最坏要枚举所有的边, 时间复杂度为 $O(mn^3)$.

构造完美匹配方案

对于 $n \times n$ 的矩阵 A, 若 $A = -A^T$, 则称 A 为斜对称矩阵, 且有:

构造完美匹配方案

对于 $n \times n$ 的矩阵 A, 若 $A = -A^T$, 则称 A 为斜对称矩阵, 且有:

- 若 n 为奇数, 则 $\det A = 0$.

构造完美匹配方案

对于 $n \times n$ 的矩阵 A, 若 $A = -A^T$, 则称 A 为斜对称矩阵, 且有:

- 若 n 为奇数, 则 $\det A = 0$.
- 若 I 是矩阵行号的一个集合, 且这些行向量构成极大线性无关组, 则 $\det A_{I,I} \neq 0$.

构造完美匹配方案

对于 $n \times n$ 的矩阵 A, 若 $A = -A^T$, 则称 A 为斜对称矩阵, 且有:

- 若 n 为奇数, 则 $\det A = 0$.
- 若 I 是矩阵行号的一个集合, 且这些行向量构成极大线性无关组, 则 $\det A_{I,I} \neq 0$.
- $rank\ A$ 必定为偶数.

构造完美匹配方案

对于 $n \times n$ 的矩阵 A, 若 $A = -A^T$, 则称 A 为斜对称矩阵, 且有:

- 若 n 为奇数, 则 $\det A = 0$.
- 若 I 是矩阵行号的一个集合, 且这些行向量构成极大线性无关组, 则 $\det A_{I,I} \neq 0$.
- $rank\ A$ 必定为偶数.
- $\det A^{i,j} \neq 0$, 当且仅当 $\det A^{\{i,j\},\{i,j\}} \neq 0$.

构造完美匹配方案

对于 $n \times n$ 的矩阵 A, 若 $A = -A^T$, 则称 A 为斜对称矩阵, 且有:

- 若 n 为奇数, 则 $\det A = 0$.
- 若 I 是矩阵行号的一个集合, 且这些行向量构成极大线性无关组, 则 $\det A_{I,I} \neq 0$.
- $rank\ A$ 必定为偶数.
- $\det A^{i,j} \neq 0$, 当且仅当 $\det A^{\{i,j\},\{i,j\}} \neq 0$.

最后一点基于 $rank\ A^{i,j} = n - 1$, 所以 $rank\ A^{\{i,j\},\{i,j\}} \geq n - 3$

构造完美匹配方案

对于 $n \times n$ 的矩阵 A, 若 $A = -A^T$，则称 A 为斜对称矩阵, 且有:

- 若 n 为奇数, 则 $\det A = 0$.
- 若 I 是矩阵行号的一个集合, 且这些行向量构成极大线性无关组, 则 $\det A_{I,I} \neq 0$.
- $rank\ A$ 必定为偶数.
- $\det A^{i,j} \neq 0$, 当且仅当 $\det A^{\{i,j\},\{i,j\}} \neq 0$.

最后一点基于 $rank\ A^{i,j} = n - 1$, 所以 $rank\ A^{\{i,j\},\{i,j\}} \geq n - 3$
又因为 $A^{\{i,j\},\{i,j\}}$ 是斜对称矩阵，秩为偶数，所以
$rank\ A^{\{i,j\},\{i,j\}} = n - 2$ 不难证明反过来也成立.

构造完美匹配方案

这个算法其实就是一个进阶版的暴力...

Algorithm 1 Matching algorithm of Rabin and Vazirani

$M = \emptyset$
for $i \in V$ do
  if $i \notin M$ then
    $compute\ \tilde{A}^{-1}(G)$
    $find\ j, such\ that\ (i,j) \in E(G)\ and\ \tilde{A}^{-1}(G)_{i,j} \neq 0$
    $G \leftarrow G - \{i,j\}$
    $M \leftarrow M \cup e_{i,j}$
  end if
end for

复杂度是感人的 $O(n^4)$.

构造完美匹配方案

我们发现上个算法的复杂度瓶颈在于每次都要重新计算 $Tutte$ 矩阵的逆矩阵，事实上，每次迭代的过程中我们只是删除了原矩阵的两行两列，却又需要重新计算 $Tutte$ 矩阵的逆矩阵，导致效率低下。

构造完美匹配方案

我们发现上个算法的复杂度瓶颈在于每次都要重新计算 $Tutte$ 矩阵的逆矩阵，事实上，每次迭代的过程中我们只是删除了原矩阵的两行两列，却又需要重新计算 $Tutte$ 矩阵的逆矩阵，导致效率低下。

那么是否存在一些性质使得我们能够每次更快地维护矩阵的逆呢？

构造完美匹配方案

我们发现上个算法的复杂度瓶颈在于每次都要重新计算 $Tutte$ 矩阵的逆矩阵，事实上，每次迭代的过程中我们只是删除了原矩阵的两行两列，却又需要重新计算 $Tutte$ 矩阵的逆矩阵，导致效率低下。

那么是否存在一些性质使得我们能够每次更快地维护矩阵的逆呢？

定理

$$A = \begin{pmatrix} a_{1,1} & v^T \\ u & B \end{pmatrix} \quad A^{-1} = \begin{pmatrix} \hat{a}_{1,1} & \hat{v}^T \\ \hat{u} & \hat{B} \end{pmatrix}$$

构造完美匹配方案

我们发现上个算法的复杂度瓶颈在于每次都要重新计算 $Tutte$ 矩阵的逆矩阵，事实上，每次迭代的过程中我们只是删除了原矩阵的两行两列，却又需要重新计算 $Tutte$ 矩阵的逆矩阵，导致效率低下。

那么是否存在一些性质使得我们能够每次更快地维护矩阵的逆呢？

定理

$$A = \begin{pmatrix} a_{1,1} & v^T \\ u & B \end{pmatrix} \quad A^{-1} = \begin{pmatrix} \hat{a}_{1,1} & \hat{v}^T \\ \hat{u} & \hat{B} \end{pmatrix}$$

$$B^{-1} = \hat{B} - \hat{u}\hat{v}^T / \hat{a}_{1,1}$$

构造完美匹配方案

简单证明

$$AA^{-1} = I_n$$

$$\begin{pmatrix} a_{1,1}\hat{a}_{1,1} + v^T\hat{u} & a_{1,1}\hat{v}^T + v^T B \\ u\hat{a}_{1,1} + B\hat{u} & u\hat{v}^T + B\hat{B} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & I_{n-1} \end{pmatrix}$$

构造完美匹配方案

简单证明

$$AA^{-1} = I_n$$

$$\begin{pmatrix} a_{1,1}\hat{a}_{1,1} + v^T\hat{u} & a_{1,1}\hat{v}^T + v^TB \\ u\hat{a}_{1,1} + B\hat{u} & u\hat{v}^T + B\hat{B} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & I_{n-1} \end{pmatrix}$$

$$\begin{aligned} I_{n-1} &= I_{n-1} - u\hat{v}^T + u\hat{v}^T \\ &= I_{n-1} - u\hat{v}^T + u\hat{a}_{1,1}\hat{v}^T/\hat{a}_{1,1} \\ &= u\hat{v}^T + B\hat{B} - u\hat{v}^T - B\hat{u}\hat{v}^T/\hat{a}_{1,1} \\ &= B(\hat{B} - \hat{u}\hat{v}^T/\hat{a}_{1,1}) \end{aligned}$$

构造完美匹配方案

简单证明

$$AA^{-1} = I_n$$

$$\begin{pmatrix} a_{1,1}\hat{a}_{1,1} + v^T\hat{u} & a_{1,1}\hat{v}^T + v^T\hat{B} \\ u\hat{a}_{1,1} + B\hat{u} & u\hat{v}^T + B\hat{B} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & I_{n-1} \end{pmatrix}$$

$$\begin{aligned} I_{n-1} &= I_{n-1} - u\hat{v}^T + u\hat{v}^T \\ &= I_{n-1} - u\hat{v}^T + u\hat{a}_{1,1}\hat{v}^T/\hat{a}_{1,1} \\ &= u\hat{v}^T + B\hat{B} - u\hat{v}^T - B\hat{u}\hat{v}^T/\hat{a}_{1,1} \\ &= B(\hat{B} - \hat{u}\hat{v}^T/\hat{a}_{1,1}) \end{aligned}$$

发现这个式子恰好对应原过程中的一次消去操作，于是问题至此解决。

最大匹配

最大匹配

首先，我们有定理：

图 G 的最大匹配数 $V(G) = \frac{1}{2} rank(G)$

最大匹配

首先，我们有定理：

图 G 的最大匹配数 $V(G) = \frac{1}{2} rank(G)$

证明应该很显然吧...

最大匹配

首先，我们有定理：

图 G 的最大匹配数 $V(G) = \frac{1}{2} rank(G)$

证明应该很显然吧...

发现本质就是找到一个最大的子图使得其具有完美匹配，而根据 $Tutte$ 定理，这样的子图必定为满秩矩阵，即一组极大线性无关组。

最大匹配

这样就可以很好的把最大匹配问题变成完美匹配问题了。

最大匹配

这样就可以很好的把最大匹配问题变成完美匹配问题了。

考虑这样的模型：

- 向原图中添加 $n-2V(G)$ 个点，并向原图的 n 个点都连边，那么这样的图一定是存在完美匹配的，就可以直接使用之前的方法求解，并且不难发现这样得到的完美匹配不考虑新加的点就是原图的最大匹配。

最大匹配

这样就可以很好的把最大匹配问题变成完美匹配问题了。

考虑这样的模型：

- 向原图中添加 $n-2V(G)$ 个点，并向原图的 n 个点都连边，那么这样的图一定是存在完美匹配的，就可以直接使用之前的方法求解，并且不难发现这样得到的完美匹配不考虑新加的点就是原图的最大匹配。
  但是这样的方法是存在一些问题的... 最坏情况下我们添加的边的数量是 $O(n)$ 的，会使得算法的常数变得很大。

最大匹配

这样就可以很好的把最大匹配问题变成完美匹配问题了。

考虑这样的模型：

- 向原图中添加 $n - 2V(G)$ 个点，并向原图的 n 个点都连边，那么这样的图一定是存在完美匹配的，就可以直接使用之前的方法求解，并且不难发现这样得到的完美匹配不考虑新加的点就是原图的最大匹配。
  但是这样的方法是存在一些问题的... 最坏情况下我们添加的边的数量是 $O(n)$ 的，会使得算法的常数变得很大。
- 其实还有另一种方法... 根据之前提到的定理，最大匹配一定等于 $\tilde{A}(G)$ 的一组极大线性无关组的完美匹配，那么我们就将其极大线性无关组求出来，然后用完美匹配方法求解即可。

最大匹配

这样就可以很好的把最大匹配问题变成完美匹配问题了。

考虑这样的模型：

- 向原图中添加 $n-2V(G)$ 个点，并向原图的 n 个点都连边，那么这样的图一定是存在完美匹配的，就可以直接使用之前的方法求解，并且不难发现这样得到的完美匹配不考虑新加的点就是原图的最大匹配。
  但是这样的方法是存在一些问题的... 最坏情况下我们添加的边的数量是 $O(n)$ 的，会使得算法的常数变得很大。
- 其实还有另一种方法... 根据之前提到的定理，最大匹配一定等于 $\tilde{A}(G)$ 的一组极大线性无关组的完美匹配，那么我们就将其极大线性无关组求出来，然后用完美匹配方法求解即可。

有没有发现最大匹配其实很水...

最大匹配

你以为最大匹配到这里就完了? Interesting?

最大匹配

你以为最大匹配到这里就完了? Interesting? Naive!

最大匹配

你以为最大匹配到这里就完了? Interesting? Naive!

显然要先观赏它的效率嘛...

最大匹配

你以为最大匹配到这里就完了? Interesting? Naive!

显然要先观赏它的效率嘛...

题目	提交者	结果	用时	内存	语言	文件大小	提交时间
#79. 一般图最大匹配	wearry	100	17631ms	4256kb	C++	3.6kb	2017-05-31 19:11:19
#79. 一般图最大匹配	wearry	100	16931ms	4232kb	C++	3.6kb	2017-05-31 19:10:10

最大匹配

你以为最大匹配到这里就完了? Interesting? Naive!

显然要先观赏它的效率嘛...

题目	提交者	结果	用时	内存	语言	文件大小	提交时间
#79. 一般图最大匹配	wearry	100	17631ms	4256kb	C++	3.6kb	2017-05-31 19:11:19
#79. 一般图最大匹配	wearry	100	16931ms	4232kb	C++	3.6kb	2017-05-31 19:10:10

其实有厉害的常数优化:

#127383	#79. 一般图最大匹配	ConstOpt	100	5346ms	7240kb	C++	6.8kb	2017-02-12 20:43:40

最大匹配

你以为最大匹配到这里就完了？Interesting？Naive!

显然要先观赏它的效率嘛...

题目	提交者	结果	用时	内存	语言	文件大小	提交时间
#79. 一般图最大匹配	wearry	100	17631ms	4256kb	C++	3.6kb	2017-05-31 19:11:19
#79. 一般图最大匹配	wearry	100	16931ms	4232kb	C++	3.6kb	2017-05-31 19:10:10

其实有厉害的常数优化:

#127383	#79. 一般图最大匹配	ConstOpt	100	5346ms	7240kb	C++	6.8kb	2017-02-12 20:43:40

然而对我来说毫无意义　（我也很绝望啊）:

#163074	#79. 一般图最大匹配	wearry	100	18189ms	4260kb	C++	7.0kb	2017-05-31 20:01:04
#163071	#79. 一般图最大匹配	wearry	100	19330ms	4176kb	C++	5.5kb	2017-05-31 19:53:53
#163064	#79. 一般图最大匹配	wearry	100	19412ms	4252kb	C++	6.1kb	2017-05-31 19:43:43
#163053	#79. 一般图最大匹配	wearry	100	17631ms	4256kb	C++	3.6kb	2017-05-31 19:11:19
#163052	#79. 一般图最大匹配	wearry	100	16931ms	4232kb	C++	3.6kb	2017-05-31 19:10:10

为了体现这种方法相较于传统的带花树算法的优势，让我们来看一道题好了。

最大匹配

Alice 和 Bob 在一个无向图上玩一个游戏，图上的某个顶点初始有一颗棋子，两人交替操作，每次可以将棋子移到一个与当前点相邻但没有访问过的点，求哪些点是先手必胜的。

最大匹配

Alice 和 Bob 在一个无向图上玩一个游戏，图上的某个顶点初始有一颗棋子，两人交替操作，每次可以将棋子移到一个与当前点相邻但没有访问过的点，求哪些点是先手必胜的。

一个显然的结论:

某个点必胜当且仅当它一定在原图的最大匹配上。

最大匹配

Alice 和 Bob 在一个无向图上玩一个游戏，图上的某个顶点初始有一颗棋子，两人交替操作，每次可以将棋子移到一个与当前点相邻但没有访问过的点，求哪些点是先手必胜的。

一个显然的结论:

某个点必胜当且仅当它一定在原图的最大匹配上。

这个问题可以用之前的第一种转化方式来处理，我们新建 $n-2*V(G)$ 个点，构造一个一定存在完美匹配的图，然后考虑点 i 能否与新加的点 k 匹配，即判断 $\tilde{A}(G)_{i,k} \neq 0$ 是否成立，若能匹配，则说明 i 不一定在最大匹配上。

最大匹配

Alice 和 Bob 在一个无向图上玩一个游戏，图上的某个顶点初始有一颗棋子，两人交替操作，每次可以将棋子移到一个与当前点相邻但没有访问过的点，求哪些点是先手必胜的。

一个显然的结论:

某个点必胜当且仅当它一定在原图的最大匹配上。

这个问题可以用之前的第一种转化方式来处理，我们新建 $n-2*V(G)$ 个点，构造一个一定存在完美匹配的图，然后考虑点 i 能否与新加的点 k 匹配，即判断 $\tilde{A}(G)_{i,k} \neq 0$ 是否成立，若能匹配，则说明 i 不一定在最大匹配上。

通过这道题目我们发现，关于一般图匹配的许多问题，使用线性代数的方法可以不需要过多的分析增广路的性质而直接利用线性代数的结论，可以减小代码以及思维复杂度。

最大匹配

再来看一个有趣的应用:

最大匹配

再来看一个有趣的应用:

Hdu 4687

给出一个无向图，求出其中哪些边一定不在最大匹配中。

$N \leq 40$

$M \leq 123$

最大匹配

再来看一个有趣的应用:

Hdu 4687

给出一个无向图，求出其中哪些边一定不在最大匹配中。

$N \leq 40$

$M \leq 123$

使用类似的方法，我们构造一个一定含有完美匹配的图，然后对于每条边，判断删除后是否有完美匹配 (类似构造方案的方法)，即可知道其是否可能在最大匹配上。

最大权匹配

最大权匹配

类似于 $Matrix-Tree$ 对带权生成树的处理，我们从 $0-W$ 枚举一个 x, 然后将每条边的边权赋值为 $x^{w_{i,j}}$ ，然后用多项式插值的方式可以得到 x 的最高次项，将其除以 2 就得到了最大权匹配的大小。

完结撒花