2015 ICPC Winter Camp Day 4
暴暴暴暴暴力

郭晓旭

上海交通大学

2015 年 2 月 4 日

Section 1

Branching

指数时间算法

什么是**指数时间算法**?

指数时间算法

什么是**指数时间算法**?
最常见的指数时间算法莫非**搜索**

最大团

Task

n 个小朋友，其中有 m 对双向的朋友关系。
求一个最大的小朋友集合，使得两两都是朋友。
$n, m \leq 200$

思考 1 分钟

解法们

请大家踊跃发言

1. $O(2^n \cdot m)$

解法们

请大家踊跃发言

2. $O(2^n \cdot n)$

解法们

请大家踊跃发言

3. $O(2^{\sqrt{2m}} \cdot nm)$

$$O(2^{\sqrt{2m}}) \cdot nm$$

因为 $\sum \deg(v) = 2m$，按照度数分类，度数 $\leq \sqrt{2m}$ 的称为小点，反之称为大点

如果最大团包含小点，设为 v_0，则最大团 $\subseteq N(v_0)$

$O(2^{\sqrt{2m}}) \cdot nm$

因为 $\sum \deg(v) = 2m$，按照度数分类，度数 $\leq \sqrt{2m}$ 的称为小点，反之称为大点
如果最大团包含小点，设为 v_0，则最大团 $\subseteq N(v_0)$
否则最大团只包含大点，而大点只有不超过 $\sqrt{2m}$ 个

最大独立集

Task

n 个小朋友，其中有 m 对双向的敌对关系
求一个最大的小朋友集合，使得两两都**不是**敌对关系

思考 1 分钟

解法们

1. $O(2^n \cdot n) = O^*(2^n)$

解法们

2. $O^*(3^{n/3})$

解法们

3. $O^*(1.3803^n)$

$O(3^{n/3})$

Algorithm $\mathtt{mis1}$**(G).**
Input: Graph $G = (V, E)$.
Output: The maximum cardinality of an independent set of G.

if $|V| = 0$ **then**
  return 0
choose a vertex v of minimum degree in G
return $1 + \max\{\mathtt{mis1}(G \setminus N[y]) : y \in N[v]\}$

Fig. 1.2 Algorithm $\mathtt{mis1}$ for MAXIMUM INDEPENDENT SET

设时间复杂度是 $T(n)$

$$
\begin{aligned}
T(n) &\le 1 + T(n - d(v_0) - 1) + \sum T(n - d(v_i) - 1) \\
&\le 1 + (d(v) + 1)\,T(n - d(v) - 1) \\
&\le 1 + s \cdot T(n - s) \\
&\le 1 + s + s^2 + \cdots + s^{n/s}
\end{aligned}
$$

当 $s = 3$ 时，右式有最大值 $O^*(3^{n/3})$

$O^*(1.3803^n)$

Algorithm `mis3`**(G).**
Input: A graph $G = (V, E)$.
Output: The maximum cardinality of an independent set of G.

1 **if** $\exists v \in V$ with $d(v) = 0$ **then**
  return $1 + \mathtt{mis3}(G \setminus v)$
2 **if** $\exists v \in V$ with $d(v) = 1$ **then**
  return $1 + \mathtt{mis3}(G \setminus N[v])$
3 **if** $\Delta(G) \geq 3$ **then**
  choose a vertex v of maximum degree in G
  return $\max(1 + \mathtt{mis3}(G \setminus N[v]), \mathtt{mis3}(G \setminus v))$
4 **if** $\Delta(G) \leq 2$ **then**
  compute $\alpha(G)$ using a polynomial time algorithm
  return $\alpha(G)$

Fig. 6.1 Algorithm `mis3` for `MIS`

设时间复杂度是 $T(n)$

$$T(n) \le T(n-1) + T(n-4)$$

解递推式得 $T(n) = O^*(1.3802^n)$

Measure & Conquer

不同度数的顶点对时间复杂度的贡献不同
不妨设图的权值是 $0.5n_2 + n_{\geq 3}$
当 $d(v) > 3$ 时，设选择/不选择 v 时，对应的权值减小为 I 和 O
对于 v 每个度数为 ≥ 3 的邻居，其权值在 I 时减少 1，在 O 时不变
对于度数为 2 的邻居，其权值在两种情况下都减少 0.5
即 $I + O \geq 2 + d(v)$，有 $T(n) \leq T(n-1) + T(n-5)$
当 $d(v) = 3$ 时，有 $T(n) \leq 2T(n-2.5)$
解得 $T(n) = O^*(1.3248^n)$

Quasiconvex Analysis

如果权值变成 $wn_2 + n_{\geq 3}$，甚至变成 $wn_2 + w'n_3 + n_{\geq 4}$，能否做得更好? 如何寻找最优的 w?

Quasiconvex Analysis

如果权值变成 $wn_2 + n_{\geq 3}$，甚至变成 $wn_2 + w'n_3 + n_{\geq 4}$，能否做得更好？如何寻找最优的 w？

当 $w = 0.596601, w' = 0.928643$ 时，$T(n) = O^*(1.2805)$

点色数

Task

n 个小朋友，其中有 m 对双向的敌对关系。
把小朋友分成最少的组，使得组内没有敌对关系

解法们

1. $O(\mathrm{Bell}(n))$

解法们

2. $O(3^n)$

解法们

3. $O((1+\sqrt[3]{3})^n)$

解法们

4. $O(2^n \cdot n)$

$O((1+\sqrt[3]{3})^n)$

设 $f(S)$ 表示 S 集合的色数，则

$$f(S) = 1 + \min_{T \subset S} f(S \setminus T)$$

$O(3^n)$?

$O((1+\sqrt[3]{3})^n)$

设 $f(S)$ 表示 S 集合的色数，则

$$f(S) = 1 + \min_{T \subset S} f(S \setminus T)$$

$O(3^n)$?
因为 T 是独立集，所以只有 $O(3^{n/3})$ 种，总的复杂度是

$$\sum_{i=0}^{n} \binom{n}{i} 3^{n/3} = (1+\sqrt[3]{3})^n$$

$O(2^n \cdot n)$

计算 c_k，表示用 k 个独立集（有序地）覆盖顶点的方案数，染色数即最小的 k 使得 $c_k > 0$
设 $f(S)$ 表示子集 S 中独立集的数量，则

$$c_k = \sum_{S \subset V} (-1)^{|S|} f(V \setminus S)^k$$

如何计算 $f(S)$? 思考 5 分钟

SPOJ TLE

逐格转移的威力

Section 2

Fixed parameter

最大独立集（再来一瓶）

Task

n 个小朋友，其中有 m 对双向的敌对关系
求一个至少 $n-k$ 的小朋友集合，使得两两都**不是**敌对关系
$n \leq 10^5, k \leq 5$

思考 5 分钟

解法们

1. $O(n^{k+1})$

解法们

2. $O(m + 3^{k(k+1)/3})$

补集转化为最小点覆盖集，设是 (V, k)
如果存在 $d(v) > k$，则归结到 $(V \setminus N[v], k-1)$
如果最终 $|V| > k(k+1)$，则无解；否则搜索

核与核化

大量问题可以参数化

解决参数化问题的常用思路是通用应用一些规则进行预处理，将问题规模限制在一个只跟 k 有关的范围内

以上过程称为核化（Kernelization），最后得到的实例称为问题的核（Kernel）

最大独立集（再来两瓶）

Task

n 个小朋友，其中有 m 对双向的敌对关系
求一个至少 k 的小朋友集合，使得两两都**不是**敌对关系
保证敌对关系是平面图

思考 1 分钟

核化

由四色定理，平面图可以 4 染色，从而存在大小至少是 $\lceil \frac{n}{4} \rceil$ 的独立集

如果 $k \le n/4$，成功

如果 $k \ge n/4$，即 $n \le 4k$，得到了大小为 $4k$ 的核

max SAT

max SAT

一个合取范式，求一个指派，使得满足至少 k 个子句

设给出的范式有 n 个变量，m 个子句

如果 $k \leq m/2$，随机一组指派，如果不能满足一半，则取反可以满足另一半

设给出的范式有 n 个变量，m 个子句
如果 $k \leq m/2$，随机一组指派，如果不能满足一半，则取反可以满足另一半
称长度超过 k 的为长子句，否则为短子句；并设长子句有 L 个。
如果 $L \geq k$，因为每次可以使用一个变量满足新的子句，所以可以被满足

设给出的范式有 n 个变量，m 个子句
如果 $k \leq m/2$，随机一组指派，如果不能满足一半，则取反可以满足另一半
称长度超过 k 的为长子句，否则为短子句；并设长子句有 L 个。
如果 $L \geq k$，因为每次可以使用一个变量满足新的子句，所以可以被满足
否则，只需在短子句中满足 $k - L$ 个即可。核的大小是 $2k^2$。

k-Path

Task

给出无向图，求一条从 s 到 t 的，长度恰好是 k 的**简单**路径

随机地给点指派 k 种颜色，对于一条长度为 k 的路径，它被正确染色的概率是

$$\frac{k!}{k^k} > \frac{k^k}{e^k k^k} = e^{-k}$$

随机 e^k 次的错误概率是

$$(1 - e^{-k})^{e^k} < (e^{-e^{-k}})^{e^k} = 1/e$$

之后怎么办？  思考 1 分钟

解法们

1. $O(k! \cdot km)$

解法们

2. $O(2^k \cdot m)$

Section 3

Method of Four Russian

二进制矩阵乘法

Task

Input $n \times n$ binary matrices A and B

Output $C = A \times B$ where

$$C_{i,j} = \sum_{1 \le k \le n} A_{i,k} B_{k,j} \pmod 2$$

An $O(n^3 / \log n)$ approach

Let $\mathbf{a}_i = A_{i,*}, \mathbf{b}_j = B_{*,j}$,

$$C_{i,j} = \langle \mathbf{a}_i, \mathbf{b}_j \rangle$$

How can we compute the (inner) product of $\mathbf{a}_i$ and $\mathbf{b}_j$ in $O(n/\log n)$ time?

Let $\mathbf{a}_i = A_{i,*}, \mathbf{b}_j = B_{*,j}$,

$$C_{i,j} = \langle \mathbf{a}_i, \mathbf{b}_j \rangle$$

How can we compute the (inner) product of $\mathbf{a}_i$ and $\mathbf{b}_j$ in $O(n/\log n)$ time?

位运算

把长度为 n 的向量 $\mathbf{v}$ 分为 $n/\log n$ 个长度为 $\log n$ 的块 $\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \ldots, \hat{\mathbf{v}}_{n/\log n}$

块的点积通过位运算 $O(1)$ 计算

$$\langle a, b\rangle = \sum_{1 \le i \le n/\log n} \mathrm{parity}(\hat{a}_i \wedge \hat{b}_i)$$

($\mathrm{parity}(x)$ counts the parity of number of 1 in x)

What's the time efficiency of parity?

Aside: parity

What's the time efficiency of parity?
To the best of my knowledge, parity can be computed in $O(\log\log n)$ with

$$\mathrm{parity}(n, w) = \mathrm{parity}(n \oplus (n \gg w), w/2)$$

总复杂度 $O(n^3 \log\log n / \log n)$

$$\langle a, b\rangle = \mathrm{parity}\left(\bigoplus_{1\le i\le n/\log n} \hat{a}_i \wedge \hat{b}_i\right)$$

gives $n^2(n/\log n + \log\log n) = O(n^3/\log n)$.

Method of Four Russian

What if without bit operations?
实际上，只有 $2^{\log n} = n$ 种不同的块，可以花费 $O(n^2 \log n)$ 的时间预处理所有块之间的点积

$$\langle a, b \rangle = \sum_{1 \le i \le n/\log n} \mathrm{table}[\hat{a}_i, \hat{b}_i]$$

And we are done.
可以优化到 $O(n^3/\log^2 n)$，留作思考

±1 Range Minimum Query

Task

Input: Array $a_1, a_2, \ldots, a_n$ where $a_i - a_{i-1} = \pm 1$, and q range queries (l_i, r_i)

Output: For each query (l_i, r_i), report the minimum element among $a_{l_i}, a_{l_i+1}, \ldots, a_{r_i}$

$O(n)$ preprocess and $O(1)$ query

Classical $O(n \log n) - O(1)$

来个人帮帮窝辣!

记 $f(i, 2^k)$ 表示 $a_i, a_{i+1}, \ldots, a_{i+2^k-1}$ 中的最小元素

$$f(i, 2^{k+1}) = \min\{f(i, 2^k), f(i + 2^k, 2^k)\}$$

$O(n \log n)$ 预处理、$O(1)$ 询问

$O(n)$ preprocess

± 1?
长度为 $(\log n/2)$ 的片段只有 $2^{\log n/2} = \sqrt{n}$ 种
询问有 $\sqrt{n}(\log n/2)^2 = O(n)$ 种

把原序列分为 $(\log n/2)$ 的小块，用 $(n/\log n)\log(n/\log n) = O(n)$ 预处理
对于询问，

- 如果跨过小块，用前缀、稀疏表、后缀回答
- 如果不跨过小块，用预处理回答

更好的方法

n 个点的不同二叉树共有

$$C_n = \frac{1}{n+1}\binom{n}{2n} \approx 4^n$$

对于不同的大小为 $(\log n/2)$ 的笛卡尔树预处理

Longest common subsequence

Task

Input 两个字符串 A, B

Output 最长公共子序列

Level Ancestor Problem

Task

Input 有根树 T 和 q 次询问 (v, d)

Output v 到根路径上深度为 d 的点

解法

树链剖分、梯子、下沉、预处理