组合数学第十讲

授课时间: 2018年11月26日　授课教师: 孙晓明　　记录人: 杨程远　陆润宇　刘卓轩

1 分拆数的上界

在第九讲中我们对分拆数$P(n)$在阶上进行了估计，证明了存在常数c_1和c_2，使得$2^{c_1\sqrt{n}} \le P(n) \le 2^{c_2\sqrt{n}\ln n}$。事实上，我们可以得到一个与下界更接近的上界：

定理 1. *存在常数c_3，使得$P(n) \le 2^{c_3\sqrt{n}}$。*

证明　上一讲中已经求出分拆数$P(n)$的生成函数为

$$P(x) = \sum_{n\ge 0} P(n)x^n = \prod_{j\ge 1}\frac{1}{1-x^j}。$$

注意到$\ln(1-x)$的Taylor展开在$x \in (-1,1)$上收敛。对$P(x)$取对数：

$$\begin{aligned}
\ln P(x) &= -\sum_{j\ge 1}\ln(1-x^j)\\
&= \sum_{j\ge 1}\sum_{k\ge 1}\frac{(x^j)^k}{k} \qquad \text{（对}\ln(1-x^j)\text{进行Taylor展开）}\\
&= \sum_{k\ge 1}\frac{1}{k}\sum_{j\ge 1}x^{jk}\\
&= \sum_{k\ge 1}\frac{x^k}{k(1-x^k)}\\
&= \sum_{k\ge 1}\frac{x^k}{k}\frac{1}{(1-x)(1+x+x^2+\cdots+x^{k-1})}\\
&\le \sum_{k\ge 1}\frac{x^k}{k}\frac{1}{(1-x)kx^{k-1}} \qquad \text{（由于}|x|<1\text{）}\\
&= \frac{x}{1-x}\sum_{k\ge 1}\frac{1}{k^2}\\
&\le \frac{2x}{1-x}。
\end{aligned}$$

最后一步中，用了$\sum_{k\ge 1}\frac{1}{k^2} = \frac{\pi^2}{6} \approx 1.645 < 2$这一事实。

另一方面，由于$\ln P(x) \ge \ln P(n)x^n = \ln P(n) + n\ln x$，我们得到了$\ln P(n) + n\ln x \le 2\cdot\frac{x}{1-x}$。于是

$$\ln P(n) \le \frac{2x}{1-x} - n\ln x。$$

令$h(x) = \frac{2x}{1-x} - n\ln x$，则只要求出$h(x)$的极小值，就能得到$\ln P(x)$的一个较好的上界。通过求导可以得到极小值点为$x = 1 - \frac{2}{n}\sqrt{2n+1} + \frac{1}{n}$。这里我们近似地取$x = 1 - \frac{1}{\sqrt{n}}$。此时，

$$\frac{2x}{1-x} \le \frac{2}{1-x} = 2\sqrt{n}。$$

而

$$\begin{aligned}
-\ln x &= -\ln\left(1-\frac{1}{\sqrt{n}}\right)\\
&= \frac{1}{\sqrt{n}}+\frac{1}{2(\sqrt{n})^2}+\frac{1}{3(\sqrt{n})^3}+\cdots\\
&\le \frac{1}{\sqrt{n}}+\frac{1}{2}\left(\frac{1}{(\sqrt{n})^2}+\frac{1}{(\sqrt{n})^3}+\cdots\right)\\
&= \frac{1}{\sqrt{n}}+\frac{1}{2}\frac{1/n}{1-1/\sqrt{n}}\\
&= \frac{1}{\sqrt{n}}+\frac{1}{2}\frac{1}{n-\sqrt{n}},
\end{aligned}$$

所以

$$-n\ln x \le \sqrt{n}+\frac{1}{2}\frac{n}{n-\sqrt{n}} \le 2\sqrt{n}。$$

最终我们有$\ln P(n) \le 4\sqrt{n}$，即

$$P(n) \le 2^{\frac{4}{\ln 2}\sqrt{n}}。$$

□

2 算法时间复杂度

例 1 快排算法的期望时间复杂度为$O(n\log n)$.

证明 我们记对n个元素进行快速排序所需的期望比较次数为$T(n)$，则由期望的定义以及快排算法，有递推式

$$T(n)=n+\frac{1}{n}\sum_{k=0}^{n-1}(T(k)+T(n-1-k))。$$

由于$T(0)=0$，有

$$T(n)=n+\frac{2}{n}\sum_{k=1}^{n-1}T(k),$$

也即

$$nT(n)=n^2+2\sum_{k=1}^{n-1}T(k)。$$

则当$n\ge 2$时，有

$$(n-1)T(n-1)=(n-1)^2+2\sum_{k=1}^{n-2}T(k)。$$

以上两式相减，即有，

$$nT(n)=(n+1)T(n-1)+2n-1,$$

从而

$$nT(n)\le (n+1)T(n-1)+2n。$$

上式两边同除以$n(n+1)$，得到

$$\frac{T(n)}{n+1}\le\frac{T(n-1)}{n}+\frac{2}{n+1}。$$

由于$T(1)=0$, 当$n\geq 2$时，有

$$\frac{T(n)}{n+1}\leq\sum_{k=2}^{n+1}\frac{2}{k}=O(\log n)。$$

因此

$$T(n)=O(n\log n),$$

即快排算法的期望时间复杂度为$O(n\log n)$。 □

例 2 3-SAT问题算法优化。

解 设逻辑表达式ϕ包含n个变量和m个子句（clause）。每个子句由3个变量或它们的非（称为文字（literal））的析取构成，整个逻辑表达式ϕ由这样的m个子句的合取组成。3-SAT问题询问是否存在一个对变量的赋值，使得这个逻辑表达式ϕ取值为真。例如，

$$\phi=(x_1\vee\neg x_2\vee x_3)\wedge(x_2\vee x_4\vee\neg x_5)$$

是一个包含5个变量、2个子句的3-SAT表达式，其中x_1,x_2,x_3,x_4,x_5的取值为$\{0,1\}$。形如$x_1=1,x_2=1,\cdots$的赋值都使得ϕ取值为真。

我们知道3-SAT问题是一个NPC问题，因此不太可能存在多项式时间的算法。另一方面，一个基本的算法是枚举所有变量的所有可能取值，代入ϕ中一一验证。若3-SAT表达式ϕ有n个变量和m个子句，则该算法的时间复杂度为$O(2^n m)$。我们希望对这个算法稍作改进。

我们每次考察当前的第一个子句，以及构成它的3个变量，枚举这3个变量的所有取值（如果不足3个就再枚举若干尚未枚举的变量，保持该层枚举变量个数为3），则显然全部8种取值中至少存在一种取值，它导致这个子句为假，从而整个逻辑表达式为假，也就是说这种取值不需要加入枚举。枚举结束后，对整个逻辑表达式进行化简，再进入下一层的枚举。设算法的最坏时间复杂度为$T(n)$，则

$$T(n)\leq 7\cdot T(n-3)+O(m)。$$

解得

$$T(n)=O(7^{\frac{n}{3}}m)=O(1.913^n m)。$$

复杂度优于之前最朴素的算法。 □

例 3 最大独立集问题（Maximum Independent Set）。

解 给定无向图$G=(V,E)$，它的一个顶点的子集$I\subseteq V$被称为图G的一个独立集，若I所包含的任意两个顶点不存在图G中的一条边。最大独立集问题要求找到图G中最大的一个独立集。这是一个NPC问题，我们同样可以枚举所有顶点的子集，然后判断得到的点集是否为独立集，最终得到一个最优解。设图G共有n个顶点和m条边，算法的时间复杂度为$O(2^n\cdot n^2)$。

我们同样想要稍作优化。设Δ为图中最大的度，那么，当$\Delta=0$或1时，该问题显然能够在多项式时间内求解。当$\Delta=2$时，该图只含有互不相交的圈、路径以及孤立点，因此仍然能够在多项式时间内求解。

下面考虑$\Delta \geq 3$的情况。在每一步，算法考虑某个度数最大的结点。如果不选择它，那么问题的规模变成$n-1$；如果选择它，那么所有与该结点相邻的结点都不能被选择，即删去了$\Delta+1 \geq 4$个结点，因此问题的规模会降至$n-4$或更少。设上述算法在最坏情况下的时间复杂度为$T(n)$，根据上述分析，我们得到$T(n) \leq T(n-1)+T(n-4)+O(m)$。

上述不等式的齐次版本对应的特征方程为$x^4 = x^3+1$，解的绝对值的最大值小于1.3803，因此

$$T(n) = O(m \cdot 1.3803^n),$$

上述算法的时间复杂度为$O(1.3803^n \cdot m)$，明显优于最朴素的算法。 □

3 专题：素数分布

定理 2. *存在无穷多个素数。*

证明 [法一] 假设只有有限个素数，记为$p_1 < p_2 < \cdots < p_n$。构造$M = p_1 p_2 \cdots p_n + 1$，则$M$是与$p_1, p_2, \cdots, p_n$均互素。那么，或者$M$本身是一个新的素数，或者$M$包含一个新的素因子，均与假设矛盾。

以上证明告诉我们，$p_{n+1} \leq p_1 p_2 \cdots p_n + 1$，因此$p_{n+1} \leq p_n! + 1$。以下我们在重复证明素数有无穷多个的同时，希望更好地估计p_{n+1}与p_n的关系。 □

[法二] 同样假设只有有限个素数，记为$p_1 < p_2 < \cdots < p_n$。考虑整数$2^{p_n}-1$，由于$2^{p_n}-1 > p_n$，根据假设，它是一个合数，即存在素数q，满足$q|2^{p_n}-1$。另一方面，根据费马小定理，$q|2^{q-1}-1$。因此$q|\gcd(2^{q-1}-1, 2^{p_n}-1) = 2^{\gcd(q-1,p_n)}-1$。若$p_n$和$q-1$互素，即$\gcd(q-1,p_n)=1$，那么$q|1$，与$q$是素数矛盾；若$p_n$和$q-1$不互素，因为$p_n$是素数，因此$p_n|q-1$，得到$q \geq p_n+1$，与$p_n$是最大的素数矛盾。综上所述，存在无穷多个素数。

上述证明说明了，$p_{n+1} \leq 2^{p_n}-1$。 □

[法三] 记费马数$F_n = 2^{2^n}+1$。观察到

$$2^{2^n}+1 = 2^{2^n}-1+2 = (2^{2^{n-1}}+1)(2^{2^{n-1}}-1)+2,$$

即$F_n - 2 = F_{n-1}(F_{n-1}-2)$。展开此递归式可以得到，$F_n - 2 = F_{n-1}F_{n-2}\cdots F_1 F_0$。因此对任意$0 \leq k \leq n-1$，$\gcd(F_n, F_k) = \gcd(F_k, 2) = 1$。因此费马数两两互素。这意味着$F_n$存在一个素因子，它不是$F_1, F_2, \cdots, F_{n-1}$的素因子，这必然要求素数有无穷多个。 □

上述证明中的法二用到的定理或引理有，

定理 3 (费马小定理)**.** *设a是一个整数，p是一个素数，且p不整除a，那么$a^{p-1} \equiv 1 \pmod p$。*

引理 4. *设a, m, n均为正整数，则$\gcd(a^n-1, a^m-1) = a^{\gcd(n,m)}-1$。*