积性函数求和的几种方法

绍兴市第一中学 任之洲

2016 年 5 月 3 日

积性函数

- 数论函数

积性函数

- 数论函数
  - 定义域为正整数，陪域为复数

积性函数

- 数论函数
  - 定义域为正整数，陪域为复数
- 积性函数

积性函数

- 数论函数
  - 定义域为正整数，陪域为复数
- 积性函数
  - 对于每一对互质的a, b均满足$f(ab) = f(a)f(b)$

积性函数

- 数论函数
  - 定义域为正整数，陪域为复数
- 积性函数
  - 对于每一对互质的a, b均满足$f(ab) = f(a)f(b)$
- 经常讨论

积性函数

- 数论函数
  - 定义域为正整数，陪域为复数
- 积性函数
  - 对于每一对互质的a, b均满足$f(ab) = f(a)f(b)$
- 经常讨论
- 如何计算？

常用计算方法

- 线性筛

常用计算方法

- 线性筛
  - $O(n)$预处理出每个数的最小质因子

常用计算方法

- 线性筛
  - $O(n)$预处理出每个数的最小质因子
  - 利用积性函数的基本性质进行递推

常用计算方法

- 线性筛
  - $O(n)$预处理出每个数的最小质因子
  - 利用积性函数的基本性质进行递推
- 杜教筛

常用计算方法

- 线性筛
  - $O(n)$预处理出每个数的最小质因子
  - 利用积性函数的基本性质进行递推
- 杜教筛
  - 能在$O(n^{\frac{2}{3}})$或$O(n^{\frac{3}{4}})$的时间内解决很多经典的数论函数求和

常用计算方法

- 线性筛
  - $O(n)$预处理出每个数的最小质因子
  - 利用积性函数的基本性质进行递推
- 杜教筛
  - 能在$O(n^{\frac{2}{3}})$或$O(n^{\frac{3}{4}})$的时间内解决很多经典的数论函数求和
  - 推导难度较高

常用计算方法

- 线性筛
  - $O(n)$预处理出每个数的最小质因子
  - 利用积性函数的基本性质进行递推
- 杜教筛
  - 能在$O(n^{\frac{2}{3}})$或$O(n^{\frac{3}{4}})$的时间内解决很多经典的数论函数求和
  - 推导难度较高
- 各有利弊

常用计算方法

- 线性筛
  - $O(n)$预处理出每个数的最小质因子
  - 利用积性函数的基本性质进行递推
- 杜教筛
  - 能在$O(n^{\frac{2}{3}})$或$O(n^{\frac{3}{4}})$的时间内解决很多经典的数论函数求和
  - 推导难度较高
- 各有利弊
- 渴求一般化的高效算法！

需要计算些什么？

需要计算些什么？

- 设$F(x)$为一个积性函数，n有k种不同质因子。

$$n=\prod_{i=1}^{k}{p_i}^{c_i}$$

$$F(n)=\prod_{i=1}^{k}F({p_i}^{c_i})$$

需要计算些什么？

- 设$F(x)$为一个积性函数，n有k种不同质因子。

$$n = \prod_{i=1}^{k} {p_i}^{c_i}$$

$$F(n) = \prod_{i=1}^{k} F({p_i}^{c_i})$$

- 对$F(x)$的前n项进行求和

$$S(n) = \sum_{i=1}^{n} F(i)$$

如何计算？

- 已知哪些信息？

如何计算？

- 已知哪些信息？
  - $F({p_i}^{c_i})$的表达式，例如

$$F({p_i}^{c_i}) = {p_i}^{c_i} + d$$

如何计算？

- 已知哪些信息？
  - $F(p_i{}^{c_i})$的表达式，例如

$$F(p_i{}^{c_i}) = p_i{}^{c_i} + d$$

  - $F(n)$为积性函数

$$F(n) = \prod_{i=1}^{k} F(p_i{}^{c_i})$$

如何计算？

- 已知哪些信息？
  - $F(p_i{}^{c_i})$的表达式，例如

$$F(p_i{}^{c_i}) = p_i{}^{c_i} + d$$

  - $F(n)$为积性函数

$$F(n) = \prod_{i=1}^{k} F(p_i{}^{c_i})$$

- 可以利用些什么？

如何计算？

- 已知哪些信息？
  - $F({p_i}^{c_i})$的表达式，例如

$$F({p_i}^{c_i}) = {p_i}^{c_i} + d$$

  - $F(n)$为积性函数

$$F(n) = \prod_{i=1}^{k} F({p_i}^{c_i})$$

- 可以利用些什么？
  - ~~函数$F(n)$的性质~~

如何计算？

- 已知哪些信息？
  - $F({p_i}^{c_i})$的表达式，例如

$$F({p_i}^{c_i}) = {p_i}^{c_i} + d$$

  - $F(n)$为积性函数

$$F(n) = \prod_{i=1}^{k} F({p_i}^{c_i})$$

- 可以利用些什么？
  - ~~函数$F(n)$的性质~~
  - 函数的积性

如何充分利用积性？

- 已知$F(n)$是一个积性函数

$$F(n)=\prod_{i=1}^{k}F({p_i}^{c_i})$$

如何充分利用积性？

- 已知$F(n)$是一个积性函数

$$F(n) = \prod_{i=1}^{k} F({p_i}^{c_i})$$

- 有什么经典算法可以参考？

如何充分利用积性？

- 已知$F(n)$是一个积性函数

$$F(n)=\prod_{i=1}^{k} F({p_i}^{c_i})$$

- 有什么经典算法可以参考？
  - 线性筛

如何充分利用积性？

- 已知$F(n)$是一个积性函数

$$F(n) = \prod_{i=1}^{k} F({p_i}^{c_i})$$

- 有什么经典算法可以参考？
  - 线性筛
- 分解质因子！

如何充分利用积性？

- 对于一个正整数n，考虑它的质因子分解

$$n = \prod_{i=1}^{k} {p_i}^{c_i}$$

如何充分利用积性？

- 对于一个正整数n，考虑它的质因子分解

$$n = \prod_{i=1}^{k} {p_i}^{c_i}$$

- n最多拥有一个大于$\sqrt{n}$的质因子

$$42 = 2 \times 3 \times \underline{7}$$

如何充分利用积性？

- 对于一个正整数n，考虑它的质因子分解

$$n = \prod_{i=1}^{k} {p_i}^{c_i}$$

- n最多拥有一个大于$\sqrt{n}$的质因子

$$42 = 2 \times 3 \times \underline{7}$$

- 在分解$1 \sim n$时，将质因子以$\sqrt{n}$为阈值分类

如何充分利用积性？

- 例如现在要分解$1 \sim 10$

$$1 \quad 2 \quad 3 \quad 4 \quad \underline{5} \quad 6 \quad \underline{7} \quad 8 \quad 9 \quad \underline{10}$$

如何充分利用积性？

- 例如现在要分解$1 \sim 10$

$$1 \quad 2 \quad 3 \quad 4 \quad \underline{5} \quad 6 \quad \underline{7} \quad 8 \quad 9 \quad \underline{10}$$

- 分为两类

$$1 \quad 2 \quad 3 \quad 4 \quad 6 \quad 8 \quad 9$$

$$1 \times \underline{5} \quad 1 \times \underline{7} \quad 2 \times \underline{5}$$

如何充分利用积性？

- 例如现在要分解$1 \sim 10$

$$1 \quad 2 \quad 3 \quad 4 \quad \underline{5} \quad 6 \quad \underline{7} \quad 8 \quad 9 \quad \underline{10}$$

- 分为两类

$$1 \quad 2 \quad 3 \quad 4 \quad 6 \quad 8 \quad 9$$

$$1 \times \underline{5} \quad 1 \times \underline{7} \quad 2 \times \underline{5}$$

- 根据积性函数值也可以分为两类计算

模型初步建立

$$\sum_{i=1}^{n} F(i) = \sum_{\substack{x \le n \\ x\text{没有大于}\sqrt{n}\text{质因子}}} F(x)\left(1 + \sum_{\substack{\sqrt{n} < p \le \lfloor \frac{n}{x} \rfloor \\ p\text{为质数}}} G(p)\right)$$

- 问题被划分为两块

模型初步建立

$$\sum_{i=1}^{n} F(i) = \sum_{\substack{x \leq n \\ x\text{没有大于}\sqrt{n}\text{质因子}}} F(x)\left(1 + \sum_{\substack{\sqrt{n} < p \leq \left\lfloor \frac{n}{x} \right\rfloor \\ p\text{为质数}}} G(p)\right)$$

- 问题被划分为两块
- 一个数x中还能加入的大质数只与$\left\lfloor \frac{n}{x} \right\rfloor$相关

模型初步建立

$$\sum_{i=1}^{n} F(i) = \sum_{\substack{x \le n \\ x\text{没有大于}\sqrt{n}\text{质因子}}} F(x) \left(1 + \sum_{\substack{\sqrt{n} < p \le \lfloor \frac{n}{x} \rfloor \\ p\text{为质数}}} G(p) \right)$$

- 问题被划分为两块
- 一个数x中还能加入的大质数只与$\left\lfloor \frac{n}{x} \right\rfloor$相关
  - 状态数只有$O(\sqrt{n})$种

模型初步建立

$$\sum_{i=1}^{n} F(i) = \sum_{\substack{x \le n \\ x\text{没有大于}\sqrt{n}\text{质因子}}} F(x) \left(1 + \sum_{\substack{\sqrt{n} < p \le \lfloor \frac{n}{x} \rfloor \\ p\text{为质数}}} G(p) \right)$$

- 问题被划分为两块
- 一个数x中还能加入的大质数只与$\left\lfloor \frac{n}{x} \right\rfloor$相关
  - 状态数只有$O(\sqrt{n})$种
- 这两个子问题都能$O\left(\frac{n^{\frac{3}{4}}}{\log n}\right)$解决

总结

- 希望利用积性函数的基本性质来优化求和

总结

- 希望利用积性函数的基本性质来优化求和
- 从分解质因数的角度，充分利用积性

总结

- 希望利用积性函数的基本性质来优化求和
- 从分解质因数的角度，充分利用积性
- 深入分析，高效计算

总结

- 希望利用积性函数的基本性质来优化求和
- 从分解质因数的角度，充分利用积性
- 深入分析，高效计算
- 不需要基于对被求和函数的推导

总结

- 希望利用积性函数的基本性质来优化求和
- 从分解质因数的角度，充分利用积性
- 深入分析，高效计算

- 不需要基于对被求和函数的推导
- 使用简单，适用范围广

感谢

- 感谢计算机协会提供学习和交流的平台。
- 感谢绍兴一中的陈合力老师、董烨华老师多年来给予的关心和指导。
- 感谢清华大学的俞鼎力、董宏华、张恒捷、王鉴浩学长对我的帮助。
- 感谢毛啸同学与我讨论这个算法，并提供了一种简洁的实现方法。
- 感谢其他对我有过帮助和启发的老师和同学。
- 感谢大家的细心聆听。