初探數論函數

Speaker 李昕威

大綱

- 前言
- 數論函數介紹
- 數論函數篩法
- 數論分塊
- 總結

前言

- 在此章節中將會講述關於數論函數的內容。
- 或許可以加個福比尼定理。

數論函數

- 定義：
- 若 $f(x)$ 是定義在正整數的函數，值域是實數或虛數。
- 則稱 $f(x)$ 是數論函數。

數論函數

- EX：
- $p(n) = \begin{cases} 1 : n \text{ 是質數} \\ 0 : n \text{ 不是質數} \end{cases}$
- 歐拉函數 $\varphi(n)$
- 因數和函數 $\sigma(n)$

積性函數

- 定義：
- 若對於任意互質的 n、m，都有 $f(nm) = f(n)f(m)$
- 則稱 $f(n)$ 是積性的。

積性函數

- $\sigma(n)$ 是積性的。
- $\sigma(n) = \prod_{i=1}^{m}(1 + p_i + \cdots + p_i^{\alpha_i})$
- 符合積性的條件。

積性函數

- 根據定義，對於任意 $n = p_1^{k_1} p_2^{k_2} \cdots p_m^{k_m}$：
- $f(n) = f\left(p_1^{k_1}\right) f\left(p_2^{k_2}\right) \cdots f\left(p_m^{k_m}\right)$
- 因此積性函數的值，可以由其在質數冪次上的取值來求出。

積性函數

-
- 艾佛森括號 $[P] = \begin{cases} 1 : \text{if } P \text{ is true} \\ 0 : \text{if } P \text{ is false} \end{cases}$
- 歐拉函數 $\varphi(n)$
- 莫比烏斯函數 $\mu(n) = \begin{cases} 1 & : n = 1 \\ (-1)^k & : n = p_1 p_2 \cdots p_k \\ 0 & : \text{Otherwise} \end{cases}$

數論函數篩法

- 給定積性函數 f，想要求 $f(1)$、$f(2)$、$\cdots$ 、$f(M)$ 的值。
- 起手式：找出 $2{\sim}M$ 中，每個數的「一個」質因數。
- 可以使用線性篩來得到。

數論函數篩法

- 多數情況下，可以（利用 DP）分成以下三種情況分析：
- 對於 n 的某個質因數 p，

1. $n = 1$
2. $p^2 \mid n$
3. Otherwise（$p \mid n$ 但 $p^2 \nmid n$）

莫比烏斯函數篩法

- $$\mu(n) = \begin{cases} 1 & : n = 1 \\ (-1)^k & : n = p_1 p_2 \cdots p_k \\ 0 & : \text{Otherwise} \end{cases}$$

- 對於 n 的某個質因數 p：

- $$\mu(n) = \begin{cases} 1 & : n = 1 \\ 0 & : p^2 \mid n \\ -\mu\left(\frac{n}{p}\right) & : \text{Otherwise} \end{cases}$$

莫比烏斯函數篩法

```cpp
vector<int> LinearMuSieve(int n, vector<int> &primeFactor) {
  vector<int> mu(n + 1, 0);
  mu[1] = 1;
  for (int i = 2; i <= n; ++i) {
    if (i % (1LL * primeFactor[i] * primeFactor[i]) == 0)
      mu[i] = 0;
    else
      mu[i] = -mu[i / primeFactor[i]];
  }
  return mu;
}
```

歐拉函數篩法

- 歐拉函數 $\varphi(n) = \prod_{i=1}^{m} p_i^{\alpha_i - 1} \times (p_i - 1)$
- 對於 n 的某個質因數 p：
- $\varphi(n) = \begin{cases} 1 & : n = 1 \\ \varphi\left(\frac{n}{p}\right) \times p & : p^2 \mid n \\ \varphi\left(\frac{n}{p}\right) \times (p - 1) & : \text{Otherwise} \end{cases}$

歐拉函數篩法

```
vector<int> LinearPhiSieve(int n, vector<int> &primeFactor) {
  vector<int> phi(n + 1, 0);
  phi[1] = 1;
  for (int i = 2; i <= n; ++i) {
    if (i % (1LL * primeFactor[i] * primeFactor[i]) == 0)
      phi[i] = phi[i / primeFactor[i]] * primeFactor[i];
    else
      phi[i] = phi[i / primeFactor[i]] * (primeFactor[i] - 1);
  }
  return phi;
}
```

數論分塊

- 這個技巧又被稱為整除分塊，~~反正都是對岸發明的詞就是了~~。
- 例題：
- 求 $\sum_{i=1}^{n} \left\lfloor \frac{n}{i} \right\rfloor$ 的值。
- $n \le 10^{10}$

數論分塊

- 對於 $i = 1 \sim \lfloor\sqrt{n}\rfloor$，$\left\lfloor\frac{n}{i}\right\rfloor$ 的取值只有 $O(\sqrt{n})$ 種。
- 對於 $i = \sqrt{n} + 1 \sim n$，$\left\lfloor\frac{n}{i}\right\rfloor$ 的取值也只有 $O(\sqrt{n})$ 種。
- 因此，不同的 $\left\lfloor\frac{n}{i}\right\rfloor$ 至多 $O(\sqrt{n})$ 種。

數論分塊

- $\sum_{i=1}^{n} \left\lfloor \frac{n}{i} \right\rfloor$ = 不同取值的數值 × 其出現的次數。
- $n = 10$ 為例：

	1	2	3	4	5	6	7	8	9	10
	10	5	3	2	2	1	1	1	1	1

數論分塊

- 定理：i 所在的塊（取值為 $\left\lfloor \frac{n}{i} \right\rfloor$）的右端點為 $\left\lfloor \frac{n}{\left\lfloor \frac{n}{i} \right\rfloor} \right\rfloor$。

	1	2	3	4	5	6	7	8	9	10
	10	5	3	2	2	1	1	1	1	1

數論分塊

- 給定 i 想要找最大的 j 使得 $\left\lfloor \frac{n}{i} \right\rfloor = \left\lfloor \frac{n}{j} \right\rfloor$，也就是說 j 是 i 所在的塊的右端點。
- $\Rightarrow \left\lfloor \frac{n}{i} \right\rfloor \le \frac{n}{j}$
- $\Rightarrow j \le \frac{n}{\left\lfloor \frac{n}{i} \right\rfloor}$
- 因為 j 是正整數，因此相當於 $j \le \left\lfloor \frac{n}{\left\lfloor \frac{n}{i} \right\rfloor} \right\rfloor$。

數論分塊

- $\sum_{i=1}^{n} \left\lfloor \frac{n}{i} \right\rfloor$ = 不同取值的數值 × 其出現的次數。
- 從 $i = 1$ 開始，計算此塊的右端點，得到出現次數。
- 把答案加進去後，將 i 改為右端點的下一個點（進入下一塊）。

數論分塊

```
for (int i = 1, r = 0; i <= n; i = r + 1) {
  r = n / (n / i);
  ans += 1LL * (r - i + 1) * (n / i);
}
```

複合數論分塊

- 求 $\sum_{i=1}^{m} \left\lfloor \frac{n}{i} \right\rfloor \times \left\lfloor \frac{m}{i} \right\rfloor$ 的值。
- $n, m \leq 10^{10}$
- 跳右端點的時候，往近的跳。

複合數論分塊

```
for (int i = 1, r = 0; i <= n; i = r + 1) {
  r = min(n / (n / i), m / (m / i));
  ans += 1LL * (r - i + 1) * (n / i) * (m / i);
}
```