

Sieve Method

如何尋找質數

- 西元前250年，希臘數學家、亞歷山大圖書館館長厄拉多塞(Eratosthese)想到了一個非常美妙的質數篩法，減少了逐一檢查每個數的的步驟，可以比較簡單的從一大堆數字之中，篩選出質數來，這方法被稱作厄拉多塞篩法(Sieve of Eratosthese)。

厄拉多塞篩法

- 問題: 找出所有小於100的質數。
 1. 留下2，刪除所有2的倍數。
 2. 留下3，刪除所有3的倍數。
 3. 留下5，刪除所有5的倍數。
 4. 留下7，刪除所有7的倍數。
 5. 留下來沒有被刪除的就是答案。

想想看

- 厄拉多塞篩法尋找100以內的質數，為何只須檢驗至7的倍數即可？

1為何不是質數？

- 現在考慮600這個正整數
∵ $600=2^3 \times 3^1 \times 5^2$ 這裡的2,3,5都是質數，此時叫做將600做了質因數分解，此種表達方式將是唯一的（順序可忽略不計），如果把1放在質數範圍內，質因數分解的表達方式便不是唯一的。
【例： $600=1^3 \times 2^3 \times 3^1 \times 5^2 = 1^5 \times 2^3 \times 3^1 \times 5^2 = 1^6 \times 2^3 \times 3^1 \times 5^2 = \dots\dots\dots$ 】

篩子法(sieve method)

1. 初始狀態

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

2. 去除2的倍數

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

3. 去除3的倍數

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	0	1	0	1	0	0	0	1	0	1	0	0	0	1	0	1	0

篩子法的程式

```
// prime.cpp
#include <stdio.h>
#define N 1000
void main()
{
    int i, j;
    bool prime[N+1];
    for(i = 2; i <= N; i++) prime[i] =
        1;
    for(i = 2; i*i <= N; i++)
        if(prime[i])
            {
                for(j = 2*i; j <= N; j++)
                    if(j % i == 0) prime[j] = 0;
            }

    for(i = 2; i < N; i++)
        if(prime[i])
            {
                printf("%4d ", i);
                if(i % 16 == 0)
                    printf("\n");
            }
    printf("\n");
}
```