5.4 節末問題 5.4 的解答

問題 5.4.1

首先，「至少出現一個 6 點」的餘事件是「全部都是 5 點以下」，因此下式會成立。

$$(\text{至少出現一個6 點的機率}) = 1 - (\text{全部都是 5 點以下的機率})$$

因此，根據乘法原理（→ **3.3.2項**）全部都是 5 點以下的機率為 $(5/6)\times(5/6)\times(5/6) = 125/216$。於是，答案如下。

$$1 - \frac{125}{216} = \frac{\mathbf{91}}{\mathbf{216}}$$

問題 5.4.2

依照 5.4.4 項的解說進行實作即可。以下為實作例。各變數和陣列的意義如下：

- gyou[i]：第 i 列的總和
- retu[j]：第 j 行的總和
- Answer[i][j]：對於第 i 列、第 j 行格子的答案

```
#include <iostream>
using namespace std;

int H, W, A[2009][2009];
int gyou[2009]; // 列的總和
int retu[2009]; // 行的總和
int Answer[2009][2009];

int main() {
	// 輸入
	cin >> H >> W;
	for (int i = 1; i <= H; i++) {
		for (int j = 1; j <= W; j++) cin >> A[i][j];
	}

	// 計算列的總和
	for (int i = 1; i <= H; i++) {
```

```
            gyou[i] = 0;
            for (int j = 1; j <= W; j++) gyou[i] += A[i][j];
        }

        // 計算行的總和
        for (int j = 1; j <= W; j++) {
            retu[j] = 0;
            for (int i = 1; i <= H; i++) retu[j] += A[i][j];
        }

        // 對各格計算答案
        for (int i = 1; i <= H; i++) {
            for (int j = 1; j <= W; j++) {
                Answer[i][j] = gyou[i] + retu[j] - A[i][j];
            }
        }

        // 以空格區隔來輸出
        for (int i = 1; i <= H; i++) {
            for (int j = 1; j <= W; j++) {
                if (j >= 2) cout << " ";
                cout << Answer[i][j];
            }
            cout << endl;
        }
        return 0;
}
```

※ Python等原始碼請參閱chap5-4.md。

問題 5.4.3 (1)

一般而言， N 以下的整數中， M 的倍數的數量為 $\lfloor N/M \rfloor$ 個， 因此：

- 3 的倍數有 $A_1 = \lfloor 1000 \div 3 \rfloor =$ **333 個**
- 5 的倍數有 $A_2 = \lfloor 1000 \div 5 \rfloor =$ **200 個**
- 7 的倍數有 $A_3 = \lfloor 1000 \div 7 \rfloor =$ **142 個**
- 15 的倍數有 $A_4 = \lfloor 1000 \div 15 \rfloor =$ **66 個**
- 21 的倍數有 $A_5 = \lfloor 1000 \div 21 \rfloor =$ **47 個**
- 35 的倍數有 $A_6 = \lfloor 1000 \div 35 \rfloor =$ **28 個**
- 105 的倍數有 $A_7 = \lfloor 1000 \div 105 \rfloor =$ **9 個**

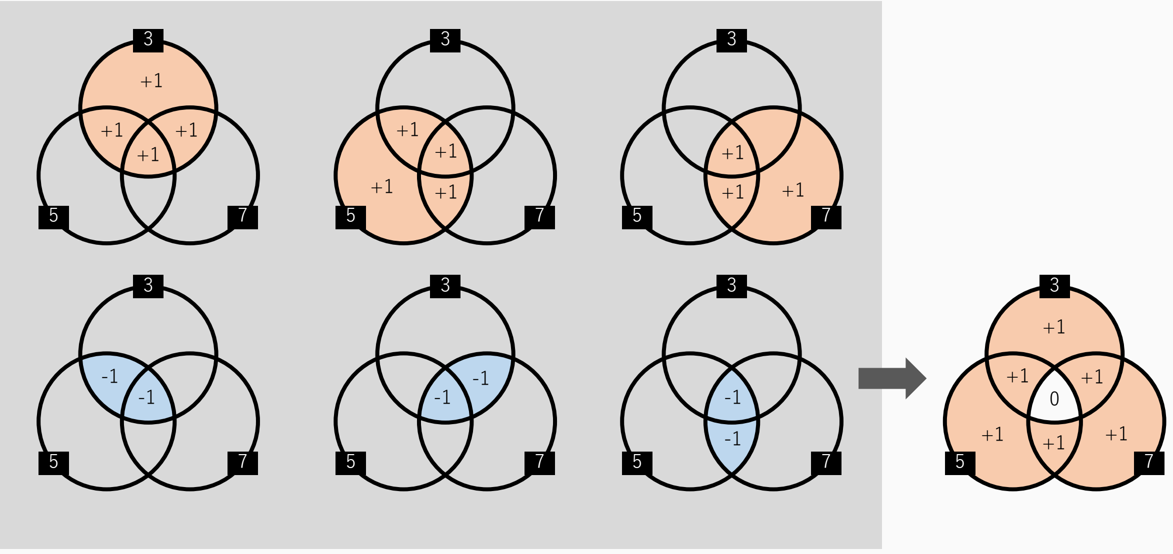
問題 5.4.3 (2), (3), (4), (5)

(2) 計算 $A_1 + A_2 + A_3$ 時，例如 **15 這個數字** 被計算了 2 次（在 A_1 和 A_2 都被算到）。

(3) 計算 $A_1 + A_2 + A_3 - A_4 - A_5 - A_6$ 時， **105 這個數字**在 A_1, A_2, A_3 被計算了 3 次，

但是在 A_4, A_5, A_6也被扣除了3次，因此完全沒有被計算到。。

如下圖所示，未正確計算到的數字只有 105 的倍數。。



(4) 將在 (3) 中未計算到的 105 的倍數加上即可。答案為 $\boldsymbol{A_1 + A_2 + A_3 - A_4 - A_5 - A_6 + A_7}$。

(5) 根據 (1) 的答案，$333 + 200 + 142 - 66 - 47 - 28 + 9 =$ **543 個**。

問題 5.4.4

集合 $S_1, S_2, S_3, \dots, S_N$ 的聯集（任一個有被包含的部分）的元素數量如下式表示。

從 N 個集合中選擇至少一個的方法有 $2^N - 1$ 種，對全部的方法加總以下的值。

- 當選擇奇數個集合時：所選集合共同部分的元素數量
- 當選擇偶數個集合時：所選集合共同部分的元素數量 $\times (-1)$

例如，集合 S_1, S_2, S_3 的聯集的元素數量為以下所有值的和。。

- S_1 的元素數量
- S_2 的元素數量
- S_3 的元素數量
- S_1 和 S_2 的共同部分 $\times(-1)$
- S_1 和 S_3 的共同部分 $\times(-1)$
- S_2 和 S_3 的共同部分 $\times(-1)$
- S_1 和 S_2 和 S_3 的共同部分 $\times(-1)$

將「1000 以下的 3 的倍數」當成 S_1，「1000 以下的 5 的倍數」當成 S_2，「1000 以下的 7 的倍數」當成 S_3 試試看。應該會是與問題 5.4.3 相同結果。

問題 5.4.5

設集合 S_1 為「N 以下 V_1 的倍數」，集合 S_2 為「N 以下 V_2 的倍數」，…，集合 S_K 為「N以下 V_K 的倍數」時，所求答案為 $S_1, S_2, \ldots, S_K$的共同部分。

因此，撰寫以下程式，將 N 個集合的選擇方法（選擇哪些倍數）進行 $2^N - 1$ 種全搜尋的話，即可得到正解。使用了位元全搜尋（→**專欄1**）這種實作方法。

此外，N 以下且為 $P_1, P_2, \ldots, P_M$ 所有值的倍數的整數個數以下式表示。

$$\frac{N}{P_1, P_2, P_3, \ldots, P_M \text{ 的最小公倍數}}$$

（3 個以上的最小公倍數的求法 →**節末問題3.2.3**）

```cpp
#include <iostream>
using namespace std;

long long N, K;
long long V[20];
long long Answer = 0;

// 回傳最大公因數的函式
long long GCD(long long A, long long B) {
    if (B == 0) return A;
    return GCD(B, A % B);
}
```

```cpp
// 回傳最小公倍數的函式
long long LCM(long long A, long long B) {
	return (A / GCD(A, B)) * B;
}

int main() {
	// 輸入
	cin >> N >> K;
	for (int i = 1; i <= K; i++) cin >> V[i];

	// 位元全搜尋
	for (int i = 1; i < (1 << K); i++) {
		long long cnt = 0; // 選擇數字的個數
		long long lcm = 1; // 最小公倍數
		for (int j = 0; j < K; j++) {
			if ((i & (1 << j)) != 0) {
				cnt += 1;
				lcm = LCM(lcm, V[j + 1]);
			}
		}
		long long num = N / lcm; // 數的個數， 此數是被選擇的所有數字的倍數
		if (cnt % 2 == 1) Answer += num;
		if (cnt % 2 == 0) Answer -= num;
	}

	// 輸出
	cout << Answer << endl;
	return 0;
}
```

※ Python等原始碼請參閱 chap5-4.md。