3.4 節末問題 3.4 の解答

問題 3.4.1

21 個のパターンすべてが等確率で起こるとは限らないからです。実際、

- $(1,1)$ の目が出る確率は $1/36$
- $(1,2)$ の目が出る確率は $1/18$

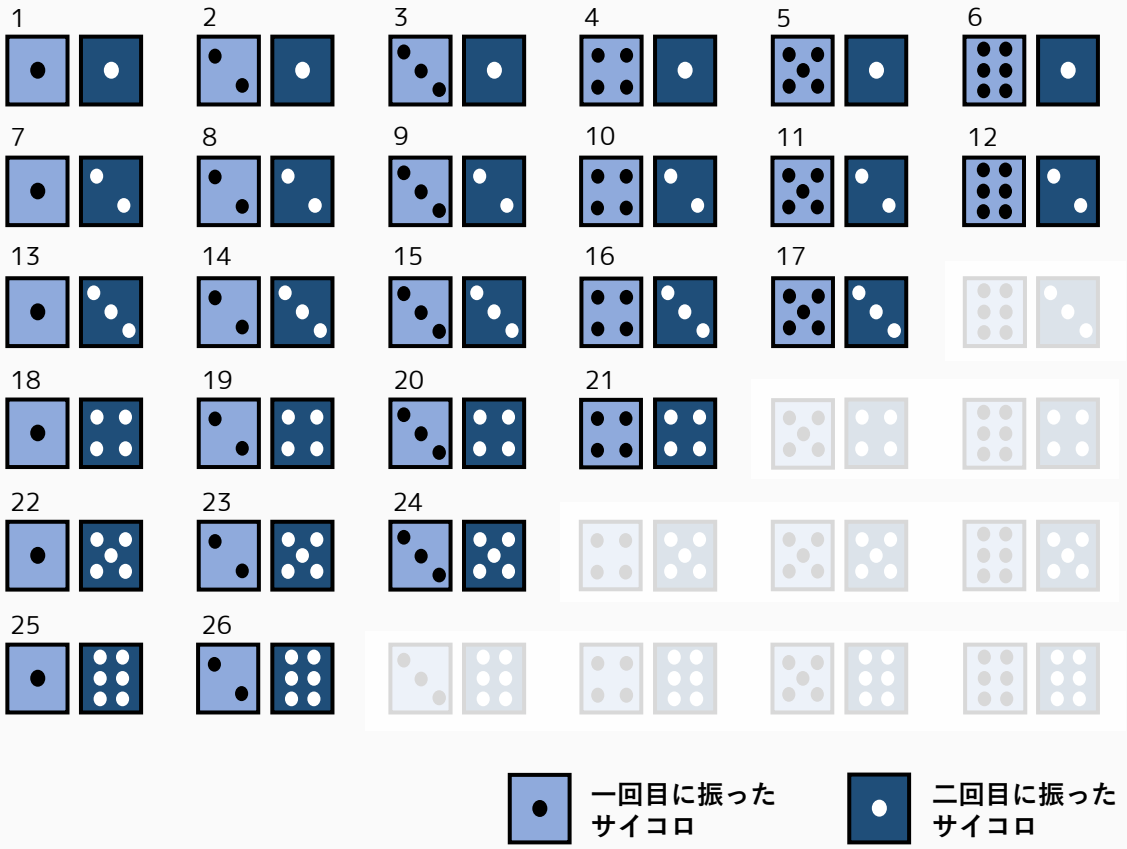
と異なるため、「21 個中 15 個のパターンで出目の和が 8 以下となるから、求める確率は $15/21$」と単純に考えることはできません。

なお、★ で示した部分は以下のようにして理解することができます。

サイコロを 2 個振ったとき、$(1,2)$ の目が出るパターンは

- 一回目に振ったものが 1 であり、二回目に振ったものが 2
- 二回目に振ったものが 2 であり、一回目に振ったものが 1

の 2 つあります。36 個中 2 個なので、確率は $2/36 = 1/18$ です。



問題 3.4.2

期待値の公式（→**3.4.2項**）に代入すると、答えは以下のようになります。

$$\left(1000000 \times \frac{1}{10000}\right) + \left(100000 \times \frac{9}{10000}\right) + \left(10000 \times \frac{9}{1000}\right) + \left(1000 \times \frac{9}{100}\right) + \left(0 \times \frac{9}{10}\right)$$

$$= 100 + 90 + 90 + 90$$

$$= 370$$

もらえる賞金の期待値が 370 円なので、参加費が 500 円の場合は損です。

問題 3.4.3

まず、期待値の線形性（→**3.4.3項**）より、次の関係が成り立ちます。

> （合計勉強時間の期待値）
> $=$（1 日目の勉強時間の期待値）$+ \cdots +$（N 日目の勉強時間の期待値）

そこで、それぞれの日の勉強時間の期待値は、以下の通りです。

- 1 日目：$(A_1 \times 1/3) + (B_1 \times 2/3)$
- 2 日目：$(A_2 \times 1/3) + (B_2 \times 2/3)$
- 3 日目：$(A_2 \times 1/3) + (B_2 \times 2/3)$
- ：
- N 日目：$(A_2 \times 1/3) + (B_2 \times 2/3)$

したがって、求める合計勉強時間の期待値は、以下のようになります。

$$\left(A_1 \times \frac{1}{3} + B_1 \times \frac{2}{3}\right) + \left(A_2 \times \frac{1}{3} + B_2 \times \frac{2}{3}\right) + \cdots + \left(A_N \times \frac{1}{3} + B_N \times \frac{2}{3}\right)$$

この値を出力するプログラムを書くと、正解となります。以下は C++ の解答例です。

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

int N;
double A[109], B[109];
double Answer = 0.0;

int main() {
```

```
    // 入力
    cin >> N;
    for (int i = 1; i <= N; i++) cin >> A[i] >> B[i];

    // 期待値を求める
    for (int i = 1; i <= N; i++) {
        double eval = A[i] * (1.0 / 3.0) + B[i] * (2.0 / 3.0);
        Answer += eval;
    }

    // 出力
    printf("%.12lf¥n", Answer);
    return 0;
}
```

※ Python などのソースコードは chap3-4.md をご覧ください。

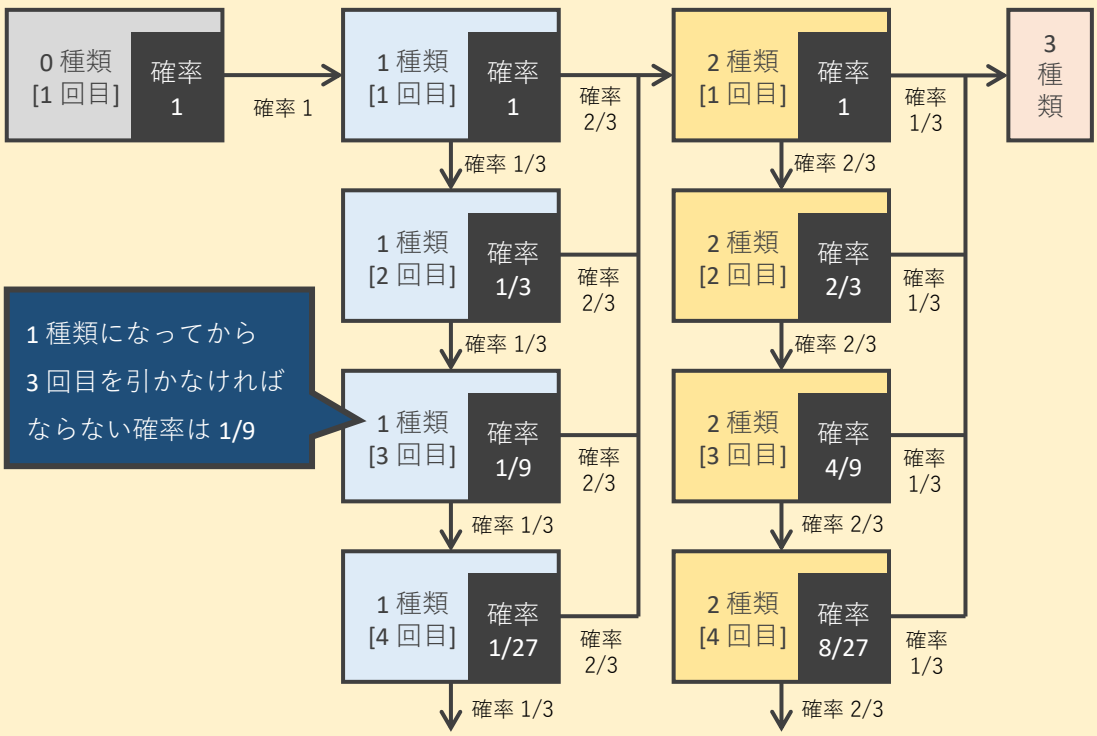
問題 3.4.4

この問題は設定が複雑で難しいので、まずは $N = 3$ の場合を考えてみましょう。

まず、全部のコインを集めるまでの過程を表した図は以下のようになります。

- 実線は、まだ集めていない種類のコインを得たこと
- 点線は、既に集めた種類のコインを引いてしまったこと

を示しています。



そこで、期待値の線形性より、全種類を集めるのにかかる回数の期待値は、

- 0 種類から 1 種類にするのにかかる回数の期待値 … (1)
- 1 種類から 2 種類にするのにかかる回数の期待値 … (2)
- 2 種類から 3 種類にするのにかかる回数の期待値 … (3)

の総和となります。

また、前ページの図と和の公式（→**2.5.10 項の最後**）より、

- (1) の期待値は 1 回
- (2) の期待値は $1+(1/3)+(1/9)+\cdots=3/2$ 回
- (3) の期待値は $1+(2/3)+(4/9)+\cdots=3$ 回

となり、$N=3$ の場合の答えは $1+3/2+3=$ **$11/2$ 回**と分かります。

次に、一般の N の場合について考えましょう。期待値の線形性より、求める答えは以下の値をすべて足したものとなります。

- 0 種類から 1 種類にするのにかかる回数の期待値
- 1 種類から 2 種類にするのにかかる回数の期待値
- ：
- $N-1$ 種類から N 種類にするのにかかる回数の期待値

そこで、すでに r 種類のコインが集まっている状態から $r+1$ 種類目のコインを集めるとき、既に集めているコインを取る確率は r/N ですから、

- 1 回以上かかる確率： 1
- 2 回以上かかる確率： $(r/N)^1$
- 3 回以上かかる確率： $(r/N)^2$
- 4 回以上かかる確率： $(r/N)^3$ [以下略]

となります。よって、回数の期待値は以下のようになります。

$$1+\left(\frac{r}{N}\right)^1+\left(\frac{r}{N}\right)^2+\left(\frac{r}{N}\right)^3+\cdots=\frac{N}{N-r}$$

最後に、全体の回数の期待値は $r=0,1,2,\ldots,N-1$ について赤い部分を足した

$$\frac{N}{N}+\frac{N}{N-1}+\frac{N}{N-2}+\cdots+\frac{N}{2}+\frac{N}{1}$$

となります。

したがって、この値を出力する以下のようなプログラムを書くと、正解が得られます。

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

int N;
double Answer = 0;

int main() {
    // 入力
    cin >> N;

    // 期待値を求める
    for (int i = N; i >= 1; i--) {
        Answer += 1.0 * N / i;
    }

    // 出力
    printf("%.12lf¥n", Answer);
    return 0;
}
```

※ Python などのソースコードは chap3-4.md をご覧ください。