整三角形の諸性質

2024/08/10 (var 0.999)

目次

補足

はじめに

カール・フリードヒッヒ・ガウスは「数学は科学の女王であり、算術は数学の女王である 」と言ったそうです。
とはいえ私にとっての算術は「ご近所の看板娘」的な存在であり、高校の頃からのお付合いがあって、仕事上の交際はあるけれども 昵懇の仲というわけではありません。その私が数論に関する資料を書いているのだから困ったものです。

ゴッドフレイ・ハロルド・ハーディ(1877 - 1945)と医師ウィルヘルム・ワインベルクによる「ハーディ＝ワインベルクの法則」のハーディがこのハーディさんだというのは長いこと知りませんでした。本人の言によると「私の最大の数学上の功績は、ラマヌジャンを見出したことである」そうですが、遠山 啓(とおやま・ひらく。1909 - 1979)さんによると、数論を通じて数学教育を行なうことの利点について以下のように述べているそうです。

「初等整数論は早期の数学教育にとってもっとも良い教材の一つであろう。それは予備知識を ほとんど必要としない。その主題は確実で、親しみやすく、用いられる推論は単純で、一般的で、また新しい。そして人間の自然な好奇心に訴える点では、数学的な学問のなかでは独自のものがある。整数論を一ヶ月うまく教えると、“技術者のための微分積分学”を一ヶ月教えたのよりも２倍も教育的で、２倍も役に立ち、１０倍もおもしろい。」

私は「初等整数論」と「プログラミング」と言換えても同じだと思っているクチですが、大人になってみたら納得はできました。　初等整数論に難関があるとすれば計算の手間がかかることですが、現代ではパソコンがあるので、プログラムが書ければかなり楽ができます。三千八百年前に数論にハマッた人々もいて、その人々が遺した数学粘土板（プリンプトン３２２とYBC 7289）をパソコンの力を借りて解読したこともあります。
ところが、数論は学校教育からは残念ながら完全にしめ出されています。小学生は整数や分数の計算はいやというほど練習させられますが、その背景にある法則に気づかせるような教え方はされてきませんでした。この状況が七十年以上前から続いています。

「『零』には四つの意味がある」などは、「空位の 0」「割り算の結果の余りとしての 0 」「空位

の０」「空っぽの０」の違いは正しく教えられません。果ては大学の理数系の学部に入ってから、解析の初歩で「有理数は稠密ではあるけれど連続ではないが、実数は稠密かつ連続である」とか言われて頭を抱えます。私は高校で「有理数は“スケスケぎっちり”で、実数は“べったり”」と教わりました。中学校でも「除算」と「割り算」の区別を教える側の先生が気にしていなかったりもします。
「除法」ではゼロで割ったらおかしなことが起きますが、「割り算」だったら余りがいくつかという話でしかありません。「5÷0＝5」と書いてあったら「法は六以上だな」と思うし「曜日の計算だよ」と言われれば「ああ、そうか」と思うだけです。
と、いうわけで、制限時間内に答えを出す一斉ペーパーテストには出題できない

- 証明された法則の適用（Plan - 計画）
- 特殊の場合についての実験（Do - 実行）
- 一般法則の推測 (Check - 測定・評価）
- 法則の証明（Action - 対策・改善）

を身につける機会に恵まれていません。これは「あらゆる科学にとって大切な思考法」なのですが。そうするとI情報処理業界に入ってくる新人が使い物にならず、早々に絶望して辞めてしまいます。それは個人にとっても社会にとっても不幸なことです。わざわざ「難問集」と表紙に書いてある参考書を買ってきて一問に一週間かけて解いたり（「三日かけて解けないくらいじゃないと“難問”とは呼べないだろ」とか嘯いていたりします）、『塵劫記』から始まって『関孝和全集』にまで手を出したりしていて、親も教師も進学先の心配なんかいていない、みたいなのがいたりするから末怖ろしいと思います。

しかるに現在の初等教育（義務教育）では、「三平方の定理」は中学校の三年生になるまで教えられません。愛知県立大学の講師である亀井 喜久男先生は、「三平方の定理は、“中学校の数学の締めくくり”に位置づけられているのではないか？」と述べていらっしゃいました。ですが、「図形の面積の加減算」は小学校ですでに教えられているので、もっと早期に、たとえば小学校で教えてもいいように思います。ユークリッドの互除法に時分で気づいちゃう小学生はけっこういます。
小学校の中学年では、すでに分数の加減算は教えられています。約分も通分も教えられているわけで、分数を教えるのと並行して「最小公倍数」「最大公約数」「ユークリッドの互除法」を教えておくのが、おそらく親切だと思います。また、中学受験では、ある数を与えて「その数の約数をすべて求めよ」などという問題も出題されることがあるので、最大公約数と最小公倍数のついでに「互いに素」「素数」「素因数分解」についても教えておいてよいのではないかと思います。これは受験にも有効です。ペーパーテストでザコ問に手間を食っていると大問を解く時間が足りなくなりますので。
三平方の定理は「ピタゴラスの定理」とかつて呼ばれていて、「ピタゴラスが発見した」とも云われていました。そうなるとヨーロッパ人は「三平方の定理の源流はギリシャにある」と言いたくなるわけですが、じつは既約ピタゴラス数の一般式を示したのはディオファントスであるらしく、古くは古代メソポタミアでも発見されていたということが確認されつつあります。だったら小学生に三平方の定理を教えるのはむしろ自然であると思われます。「もう、環とか剰余系とかもいけるんじゃねぇ？」と思います。
ソフトウエア業界にいると、このあたりの問題は切実です。プログラミング講習の第一歩は“Hellow, World!”で、二歩めが「ユークリッドの互除法」と相場が決まっているのですが、「『ユークリッドの互除法のプログラムを書け』というと六割が頭を抱える」と零したら「おまえのところの新人は優秀だな。うちの会社では八割を超える。しかもそのうちの半分はすぐ辞める」という真面目な話がありました。もう数学を受けいれられない身体になっています。

直角三角形の性質は、おそらくは古代エジプトやシュメールの時代にはけっこう知られていて、土木や建築の分野では実学でした。それが近年になって分かってきたので、「ピタゴラスの定理」も「三平方の定理」と改称されたようです。
ところが二十一世紀になって、「古代メソポタミアの数学粘土板の双璧」と云われる二枚の粘土板をパソコンを使って解読してみたところ、古代メソポタミアの数学は意外に高度なものだったということが判りました。ただ、メディアが粘土板なので、結果だけ書いてあって証明がないというのが困りものです。シュメールの時代から書記の学校はあったので、授業では教わっていたと思われます。
さらに、その成果は(少なくとも三平方の定理に関しては)七世紀インドのブラフマグプタや十七世

紀のピエール・ド・フェルマーにも先んじていて、二十世紀に証明された B=H=K 定理にまで及んでいました。

ですが、こういう面白い話は、学校や学習塾や予備校では嫌われます。「受験対策にならない」と思われているからです。
とはいえ、「受験戦争」と呼ばれた時代にも、高木貞治(1875 - 1960)『初等整数論講義』や遠山 啓『初等整数論』『現代数学の考え方』やらをフツーに読んでいた高校生はフツーにいました。
受験数学では「他の人に訊いてはいけない」「資料持込不可」「電卓は使ってはならない」「論述式の問題が少ない」など、いろいろな制約があって面白くありませんが、基本的に「なんでもあり」です。
小平 邦彦先生には『ボクは算数しか出来なかった』という自伝があります。矢野 健太郎先生は『お母さまのさんすう 新版』『数学の考え方』を著しています。一松 信先生には『教室に電卓を！』『整数とあそぼう』などがあります。とにかく数学にはあっちゃこっちゃに門があり、間口は広いし敷居は低いし、壁なんかあって無きがごとしの世界で、しかもその庭は「広大な遊び場」です。算数や数学のテストように、「他の人に答えを訊いてはいけない」とか「資料持込不可」とかいったややこしい制約はありません。

そこで、数学者の遠山 啓さんによる「タイルのシェーマ」と十ミリ方眼紙と電卓とパソコンを駆使して、既約ピタゴラス数と既約ピタゴラス三角形を代表とする各種の三角形の性質の解明に挑んでみようというのが本文書の目標です。「数論とパソコンと工学では範囲が広すぎて子供が迷ってしまうのではないだろうか？」という心配はご無用です。親切な案内書もありますので。

用語について

- （正の整数という教義の）自然数 {x, y, z} について、$x^2 + y^2 = z^2$ が成り立つものを**ピタゴラス数**と呼びます。
- 辺の長さがピタゴラス数である三角形を**ピタゴラス三角形**と呼びます。
- すべての相似なピタゴラス三角形のうち、最小のものを**既約ピタゴラス三角形**と呼びます。ただし、一般的には「既約ピタゴラス三角形」ではなく「原始ピタゴラス三角形」というのが一般的です。
- ピタゴラス三角形において、「直角をはさむ二辺」を英語で“leg”といいます。日本語に訳すと「脚」ですが、一般的には「足（英語でいうと foot とか feet でしょうか？）」が使われますが、本文書では「脚」で統一します。
- 既約ピタゴラス三角形の辺長を**既約ピタゴラス数**といいます。なお、既約ピタゴラス三角形の辺長は、{3, 4, 5} とか {4, 3, 5} とか書くのですが、本文書では{奇数, 偶数, 残りのいちばん長い斜辺の長さ(これは必ず奇数です)}の順番、すなわち {o(odd。奇数), e(even。偶数), h(hypotenuse)} の順番で書くことにします。奇数辺と偶数辺のどちらが長いかが分かっているときには、ときどき {a, b, c}（$a < b < c$）と書く流儀があります。
- これは筆者の造語ですが、合同な既約ピタゴラス三角形をふたつくっつけた長方形を**既約バビロニア長方形**と呼ぶことにします。説明のうえでは便利です。こちらは {o(odd。奇数), e(even。偶数), d(diagonal。対角線)} と表記します。最初に「原始バビロニア長方形」として広めてしまったので、ネットで検索すると「原始バビロニア長方形」がヒットします。

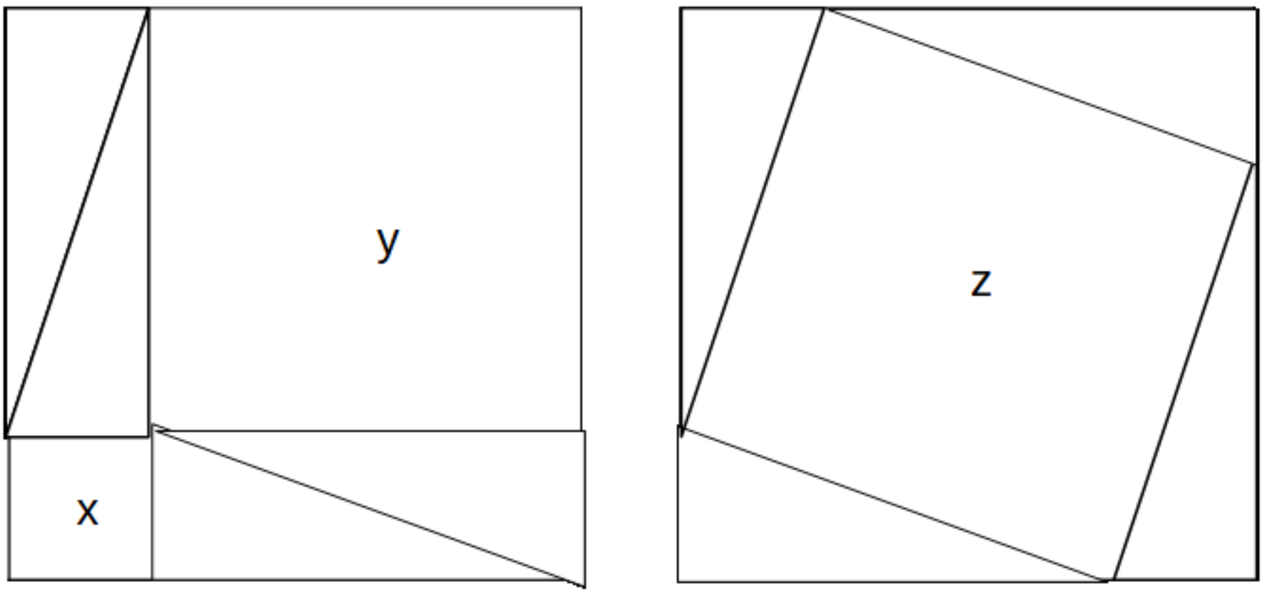
第 1 章：既約ピタゴラス数の性質

とりあえず、学校で教わるように直角三角形を想像してください。このとき「直角をはさむ二本の辺の長さがともに百以下の自然数」であるものを考えましょう。鏡像対称のものを別扱いすると、

100 × 100 で一万個あります。

「いわゆるユークリッド平面上の三角形におけるある辺の長さは、他の二辺の長さの和よりも短い」ということは既知としましょう。ここで、「直角三角形の三本の辺の長さがすべて自然数であるもの」は、この中にいくつあるでしょうか？

ただし、三平方の定理は以下のような図形から自明であることにしましょう。



こうなると「$1 \leqq x \leqq 100$」かつ「$1 \leqq y \leqq 100$」かつ「$^2 + y^2 = z^2$」が成立する場合を見つければいいわけです。すなわち $1 < z < (x + y)$ の範囲で z を探せばいい。

全部で一万通りあるケースをぜんぶチェックするのは、人間にとっては大変ですが、コンピュータで探すぶんにはそれほど時間はかからないはずです。少なくとも一晩ぶん回せば答えは出そうです。なるべく素朴な方法で、この値を計算してみましょう。コードはこれ。

※ただしメソッドの定義だけなので、実際に動くまるごとのクラス定義と差し替える予定です。

```
/**
 *
 */
    static final int UPPER_LIMIT = 100;

    public static void main( String [] args ) {
        // _main1();

        // ピタゴラス数の数
        _main2();

        // x < y < z の数
        _main3();

        // 既約ピタゴラス数の数
        _main4();

    }

    /**
     * 長方形の個数（縦横は区別する）
     */
    public static void _main1() {
        int cnt = 0;
        int cntx = 0;
        Chronograph.start();
```

```
        for (int x= 1; x <= UPPER_LIMIT; x += 1) {
            for (int y = 1;  y <= UPPER_LIMIT; y += 1) {
                cnt += 1;
            /*
                int cc = o * o + e * e;
                for (int d = 1; d < (o + e); d += 2) {
                    if ((cc == d * d) && (abax.Abaci.gcf(e,o) == 1) ) {
                        if ( d < 1000 ) {
                            cnt += 1;
                            if (o < e) {
                                System.out.print("*");
                                cntx += 1;
                            }
                            System.out.println("{" + o  + ", " + e + ", " + d + "}");
                            break;
                        }
                    }
                }
                */
            }
        }
        System.out.println();
        System.out.println(Chronograph.stop() + " msec.");
        System.out.println(cnt);
        System.out.println(cntx);
    }

    /**
     * うちピタゴラス数の数
     *
     */

    public static void _main2() {
        int cnt = 0;
        int cntx = 0;
        Chronograph.start();
        for (int x = 1; x <= UPPER_LIMIT; x += 1) {
            for (int y = 1;  y <= UPPER_LIMIT; y += 1) {
                int dd = x * x + y * y;

                for (int d = 1; d < (x + y); d += 1) {
                //  if ((dd == d * d) && (abax.Abaci.gcf(e,o) == 1) ) {
                        if ( d * d == dd ) {
                            cnt += 1;
                            /*

                            if (o < e) {
                                System.out.print("*");
                                cntx += 1;
                            }
                            */
                            System.out.print("{" + x  + ", " + y + ", " + d + "}, ");
                            cntx += 1;
                            if ((cntx % 10) == 0) {
                                System.out.println();
                            }
                            break;

                        }
                    }
                }
            }
    System.out.println(cntx);
    System.out.println(Chronograph.stop() + " msec.");
    System.out.println(cnt);
    System.out.println(cntx);
    }

    /**
     * 既約ピタゴラス数。
```

```
     *
     *
     */
    public static void _main3() {
        int cnt = 0;
        int cntx = 0;
        Chronograph.start();
        for (int x= 1; x <= UPPER_LIMIT; x += 1) {
            for (int y = x + 1;  y <= UPPER_LIMIT; y += 1) {

                int dd = x * x + y * y;

                for (int d = 1; d < (x + y); d += 1) {
                        if ((dd == d * d) && (abax.Abaci.gcf(x, y) == 1) ) {
                            cnt += 1;
                            // System.out.print("*");
                                cntx += 1;
                                System.out.print("{" + x  + ", " + y + ", " + d + "}, ");
                                if ((cntx % 10) == 0) {
                                    System.out.println();
                                }

                            break;
                        }
                    }
                }
            }
    System.out.println();
    System.out.println(Chronograph.stop() + " msec.");
    System.out.println(cnt);
    System.out.println(cntx);
    }

    public static void _main4() {
        int cnt = 0;
        int cntx = 0;
        Chronograph.start();
        for (int o = 1; o < UPPER_LIMIT; o += 2) {
            for (int e = 2;  e <= UPPER_LIMIT; e += 2) {
//              cnt += 1;
                int dd = o * o + e * e;

                for (int d = 1; d < (o + e); d += 1) {
                        if ((dd == d * d) && (abax.Abaci.gcf(o, e) == 1) ) {
                            /*
                            if (o < e) {
                                System.out.print("*");
                                cntx += 1;
                            }
                            */
                            cnt += 1;
                            System.out.println("{" + o  + ", " + e + ", " + d + "}");
                            break;
                        }
                    }
                }
            }
    System.out.println();
    System.out.println(Chronograph.stop() + " msec.");
    System.out.println(cnt);
    System.out.println(cntx);
    }


    public static void _main() {
        int cnt = 0;
        int cntx = 0;
        Chronograph.start();
        for (int o = 1; o < UPPER_LIMIT; o += 2) {
            for (int e = 2;  e < UPPER_LIMIT; e += 2) {
```

```
                int cc = o * o + e * e;
                for (int d = 1; d < (o + e); d += 2) {
                    if ((cc == d * d) && (abax.Abaci.gcf(e,o) == 1) ) {
                        if ( d < 1000 ) {
                            cnt += 1;
                            if (o < e) {
                                System.out.print("*");
                                cntx += 1;
                            }
                            System.out.println("{" + o  + ", " + e + ", " + d + "}");
                            break;
                        }
                    }
                }
            }
        }
        System.out.println();
        System.out.println(Chronograph.stop() + " msec.");
        System.out.println(cnt);
        System.out.println(cntx);
    }
}
```

で、結果はこれです。

{3, 4, 5}, {4, 3, 5}, {5, 12, 13}, {6, 8, 10}, {7, 24, 25}, {8, 6, 10}, {8, 15, 17}, {9, 12, 15}, {9, 40, 41}, {10, 24, 26},
{11, 60, 61}, {12, 5, 13}, {12, 9, 15}, {12, 16, 20}, {12, 35, 37}, {13, 84, 85}, {14, 48, 50}, {15, 8, 17}, {15, 20, 25}, {15, 36, 39},
{16, 12, 20}, {16, 30, 34}, {16, 63, 65}, {18, 24, 30}, {18, 80, 82}, {20, 15, 25}, {20, 21, 29}, {20, 48, 52}, {20, 99, 101}, {21, 20, 29},
{21, 28, 35}, {21, 72, 75}, {24, 7, 25}, {24, 10, 26}, {24, 18, 30}, {24, 32, 40}, {24, 45, 51}, {24, 70, 74}, {25, 60, 65}, {27, 36, 45},
{28, 21, 35}, {28, 45, 53}, {28, 96, 100}, {30, 16, 34}, {30, 40, 50}, {30, 72, 78}, {32, 24, 40}, {32, 60, 68}, {33, 44, 55}, {33, 56, 65},
{35, 12, 37}, {35, 84, 91}, {36, 15, 39}, {36, 27, 45}, {36, 48, 60}, {36, 77, 85}, {39, 52, 65}, {39, 80, 89}, {40, 9, 41}, {40, 30, 50},
{40, 42, 58}, {40, 75, 85}, {40, 96, 104}, {42, 40, 58}, {42, 56, 70}, {44, 33, 55}, {45, 24, 51}, {45, 28, 53}, {45, 60, 75}, {48, 14, 50},
{48, 20, 52}, {48, 36, 60}, {48, 55, 73}, {48, 64, 80}, {48, 90, 102}, {51, 68, 85}, {52, 39, 65}, {54, 72, 90}, {55, 48, 73}, {56, 33, 65},
{56, 42, 70}, {56, 90, 106}, {57, 76, 95}, {60, 11, 61}, {60, 25, 65}, {60, 32, 68}, {60, 45, 75}, {60, 63, 87}, {60, 80, 100}, {60, 91, 109},
{63, 16, 65}, {63, 60, 87}, {63, 84, 105}, {64, 48, 80}, {65, 72, 97}, {66, 88, 110}, {68, 51, 85}, {69, 92, 115}, {70, 24, 74}, {72, 21, 75},
{72, 30, 78}, {72, 54, 90}, {72, 65, 97}, {72, 96, 120}, {75, 40, 85}, {75, 100, 125}, {76, 57, 95}, {77, 36, 85}, {80, 18, 82}, {80, 39, 89},
{80, 60, 100}, {80, 84, 116}, {84, 13, 85}, {84, 35, 91}, {84, 63, 105}, {84, 80, 116}, {88, 66, 110}, {90, 48, 102}, {90, 56, 106}, {91, 60, 109},
{92, 69, 115}, {96, 28, 100}, {96, 40, 104}, {96, 72, 120}, {99, 20, 101}, {100, 75, 125}

一万個あったものが 126 個まで減りました。1.26 %です。私のパソコンでは 計算時間は 0.109 秒かかりました。佛教語の「刹那」は七十五分の一秒くらいだそうですから、八刹那よりも長いくらいです。一晩どころかトイレに行くヒマもありません

ところが、この中には、「より小さい直角三角形と相似であるもの」も含まれています。それを除いて、「相似のもののうち、いちばん小さいもの」だけを遺してみましょう。こうなります。

{3, 4, 5}, {5, 12, 13}, {7, 24, 25}, {8, 15, 17}, {9, 40, 41}, {11, 60, 61}, {12, 35, 37}, {13, 84, 85}, {16, 63, 65}, {20, 21, 29}, {20, 99, 101}, {28, 45, 53}, {33, 56, 65}, {36, 77, 85}, {39, 80, 89}, {48, 55, 73}, {60, 91, 109}, {65, 72, 97}

　これで 18 個 まで減りました。計算時間は 0.078秒。
　これを見ると、「斜辺の長さは奇数」「直角をはさむ二辺は、一方は奇数、一方は偶数」でありそうだ、という見当がつきます。
　そんなわけで、これを{奇数辺, 偶数辺, 斜辺} の順とし、これを {o, e, d} と置いて整理すると、こんなことになります。

{3, 4, 5}
{5, 12, 13}
{7, 24, 25}
{9, 40, 41}
{11, 60, 61}
{13, 84, 85}
{15, 8, 17}
{21, 20, 29}
{33, 56, 65}
{35, 12, 37}
{39, 80, 89}
{45, 28, 53}
{55, 48, 73}
{63, 16, 65}
{65, 72, 97}
{77, 36, 85}
{91, 60, 109}
{99, 20, 101}

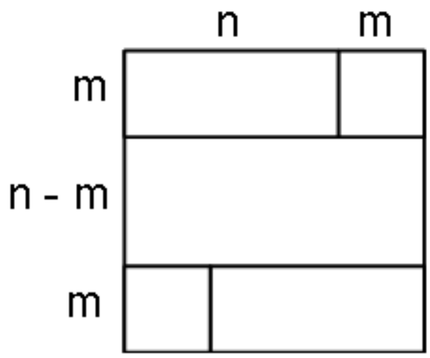
　これだと奇数が 50 個に偶数が 50 個で計 2500 個です。ただしユークリッドの互除法を使うので、計算時間については確かなことは謂えません。
　これは、三角形として見れば既約ピタゴラス三角形で、数で考えると既約ピタゴラス数です。どちらで考えても、もちろん一対一対応します。
　この偶数脚の辺長は、4, 8, 12, 16, 20, 24, 28, 36, 40, 48, 56, 60, 72, 80, 84, ……と、なんとなく 4 の倍数のような気がしてきます。そこで、

- 証明された法則の適用（Plan - 計画）
- 特殊の場合についての実験（Do - 実行）
- 一般法則の推測 (Check - 測定・評価)
- 法則の証明（Action - 対策・改善）

という PDCA サイクルの発動です。
　なけなしの知識のなかの「証明された法則の適用」を行なって、パソコンで「特殊の場合についての実験」を行ないます。それで「これは証明できそうだ」と思ったら「一般法則の推測」を行ないます。「法則の証明」は数学者とかに任せるなりなんなりすればいいんですが、なかなか頼りになる人は見つかりません。そこが困ったところです。
※この証明はのちほど。

第 2 章：既約ピタゴラス数の一般式

　原始ピタゴラス数の一般式は、少なくともユークリッドの時代には知られていました。おそらくはこのような図形から求められたようです。



これは、$(m + n)\times(m + n)$ の正方形のなかに $m \times n$ の長方形を二つ入れると、$m \times n$ の長方形二つと $m \times m$ の正方形二つの間に、$((m + n) - 2m)\times(m + n)$ の長方形の隙間ができる、ということです。つまり、 $m + n < 2n$ でないと困ります。ここから $m < n$ という条件件が出てきます。さらによく見ると、「これは m と n が互いに素でないと具合悪いな」というのがなんとなく感じられます。

このとき、この隙間が原始ピタゴラス数の奇数項 o (間の隙間部分) に、面積 m^2の正方形二個分を足すと、斜辺 h に一致すると気づいたようです。これを整理すると、

$$\{o, e, h\} = \{((m + n) - 2m)\times(m + n) , 2mn, ((m + n) - 2m)\times(m + n) + 2m^2\}$$

となり、これは

$$\{o, e, h\} = \{n^2 - m^2, 2mn, m^2 + n^2\}$$

となります。

はい、ユークリッドの一般式の一丁上がりです。

ただ、この一般式は「隙間にあたる $n^2 - m^2$ を二回使っているところが美しくない」とは思うんですよ。他にも、m^2の正方形と $m \times n$ の長方形が二つづつ出てくるとか「m と n の偶奇が異なる」というのもなんとなく「美しくねぇなぁ」と思います。

ここまでは、「直角三角形」「ピタゴラスの定理」「既約ピタゴラス数」を基礎とした議論でした。図形的には「三辺の長さがすべて自然数であり、互いに素である直角三角形」を考えています。このあたりはユークリッド流です。ただし、「もうちょっと美しい図はないものだろうか？」と思った人もいたようです。

たとえば、七世紀のインドのブラフマグプタが書き残した以下の式があります。

ブラフマグプタによる既約ピタゴラス数の一般式

- $\{o, e, d\} \equiv \{ pq, \frac{q^2 - p^2}{2} , \frac{p^2 + q^2}{2} \}$
- ただし、p と q はともに奇数であり、互いに素である。ここでは $p < q$ とする。

ここから [p, q] を求めるには、

$$e + d = \frac{q^2 - p^2}{2} + \frac{p^2 + q^2}{2} = q^2$$

$$\therefore q = \sqrt{q^2}$$

$$o = pq$$

$$p = \frac{o}{q}$$

を使う。

ブラフマグプタはあまり知られていませんが、「零の発見者」と言われています。すなわち、昭和十四年（1934）に著された吉田洋一『零の発見 — 数学の生いたち』（岩波新書）では、

- いかなる数に零を乗じても結果はつねに零であること
- いかなる数に零を加減してもその数の値に変化がおこらないこと

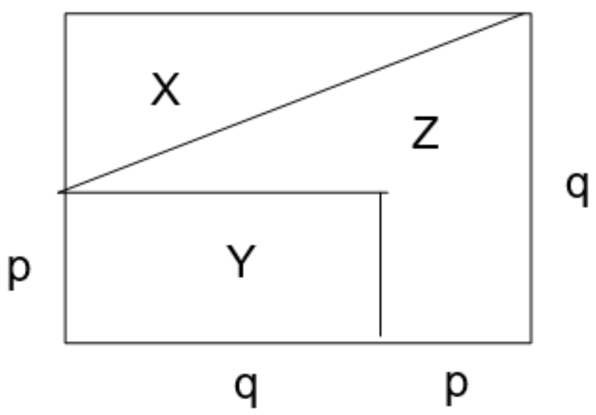
ということを書きのこしたのがブラフマグプタだと書かれています。

「０」には

- 空位を表す０。一般的には「とんで」が使われる。1024は「一千とんで二十四（ふたじゅうよん）・九九では「が」が用いられ、$2 \times 3 = 6$ は $2 \times 3 = 06$ なので「にさんがろく」
- 「割ったときの余りが０（割り切れる）」の０。
- 数としての０。
- 「空っぽ」の０。後に別概念とされ、アンドレ・ヴァイユ（André Weil）がスカンジナビア文字「Ø」から採って命名したと自伝に書いている。箱は基本的に何を入れてもいいので、茶箱を非常食を入れても乾燥食品を入れても一向にかまわない。数学では「要素の内包的定義がないので、すべつの空集合は同一である」ということになっているが、このあたりを疎かにするプロも多いという話は大御所の先生から聞いた。

という四つの「0」があるわけですが、(割り算ではなく除算の世界で)「『÷0』を考えるとわけのわからないとことが起きる」（そのせいで、インドでは０は「悪魔の数字」と云われたそうです）というところま で０の正体を追いつめたのがブラフマグプタであるようです。

ともあれ、{o, e, h} として、ブラマグプタの公式であらわせます。七世紀のインドの数学がどのようなものであったかは知らないのですが、[p, q] は以下のような図で表されたのかもしれません。



なお、[m, n] と [p, q] の間には

$m = \frac{q - p}{2}$

$n = \frac{p + q}{2}$

という関係があります。つまり、逆は

$p = n - m$

$q = m + n$

です。

第３章：インドの数学

[m, n] の初項を [1, 2]とし、次の項を [m', n'] = [n, 2n + m] とすると、
[1, 2], [2, 5], [5, 12], [12, 29], [29, 70], [70, 169], [169, 408], [408, 985], [985, 2378], [2378, 5741], ……
となります。
「それが何だというのか？」という感じがしますが、ここから既約ピタゴラス数 {o, e, h} を求めると、

{3, 4, 5}, {21, 20, 29}, 119, 120, 169}, {697, 696, 985}, {4095, 4060, 5741}, {23661, 23662, 33461}, {137903, 137904, 19525}, ……

となって、既約ピタゴラス三角形がどんどん直角二等辺三角形に近づいてゆきます。
これを使って２の平方根の近似値を求めると、7 項めの [169, 408] くらいで「ヒトヨヒトヨニヒトミゴロ」まで出てきます。
ここに出てくる数値を大きさの順に並べると、

1, 2, 5, 12, 29, 70, 169, 408, 985, 2378, 5741 ……

となって、これはペル数というものになります。
すなわち、
$B_0 = 1$
$B_1 = 2$
として、
$B_n = 2B_{n-1} + B_{\mathrm{n-2}}$
です。
このペル数は、ブラフマグプタの弟子筋にあたるバースカラ2世が、ディオファントス方程式の解法の研究中の 1150 年に発見したそうです。

ちなみに、同じことを [p, q] でどうなるかというと、
<@> [1, 3]、[3, 7]、[7, 17]、[17, 41]、[41, 99]、[99, 239]、[239, 577]、[577, 1393]、[1393, 3363]、[3363, 8119] ……

から
1, 3, 7, 17, 41, 99, 239, 577, 1393, 3363, 8119 ……
が出てくるのですが、なんかしらこの数列は $\tan^{-1}()$ かなんかの計算のときに見かけた気はしますが、これを追いかけていると話が終わらなくなるので省略します。

第 4 章：各種の証明

既約ピタゴラス数の基本的な性質には、以下のようなものがあります。

- 既約ピタゴラス数は無数に存在する。
- {o, e, h} のうち、o と h は奇数であり、e は偶数である。
- e は 4 の倍数である。
- 最小（面積が最小、かつ周長が最小）の既約ピタゴラス数は、{3, 4, 5} である。これはユークリッドの公式の [1, 2]、ブラフマグプタの公式の [1, 3] に対応する。
- 既約ピタゴラス数の積は 60 の倍数である。
- 直角二等辺三角形の辺長は既約ピタゴラス数ではない。

ひとつづつ証明してゆきましょう。

- 「既約ピタゴラス数は無数に存在する」は、ブラフマグプタの式は「互いに素である奇数 p と q（ただし p < q）が無数に存在することから謂えます。

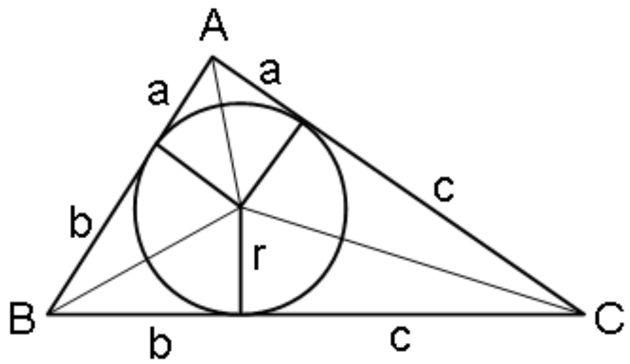
- 「{o, e, d} のうち、o と d は奇数であり、e は偶数である。」と「e は 4 の倍数である」と「o × e × d は六十の倍数である。」はまとめて説明したほうが理解しやすそうなので、そうします。
  「pq が奇数」というのは、「奇数×奇数は奇数である」ことからわかります。
  「d が奇数」というのは「奇数×奇数は奇数である」ので p^2も q^2 も奇数で、その差は偶数でなので 2 で割れる …… まではいいんですが、 7 - 3 は 4 なので 2 で割っても偶数であって、これだけでは証明になりません。
  ここで、「o と e と d が互いに素」というのが利いてきます。e は必ず偶数なので、e と d が両方とも偶数だったら最大公約数 2 が出てきて「e と d が互いに素」という条件が満たされません。そこで「e が偶数で、しかも 4 の倍数である」ほうから攻めてゆきましょう。
  そうなると、「q^2 - q^2が 8 の倍数である」というのがとっかかりになりそうです。こうなるとプログラマは強いんです。
  任意の奇数 Ω は mod 8 で考えると、{1, 3, 5, 7} のどれかということになります。そうすると p^2 も q^2 も 1(mod 8) になるので、差をとると 0 になります。0 - 0 = 0 なので、「奇数の平方の差は 8 で割切れる」ことがいえます。そうなると、「(p^2 + 2pq + q^2) / 2」が奇数というのもなんかしら証明できそうな気がしませんか？　ただし、こうなると剰余系とかが出てきそうで、「中学生にもわかる証明」となると見つけられずに悩んでいます。
  つぎは「最小（面積が最小、かつ周長が最小）の既約ピタゴラス数は、{3, 4, 5} である」です。これはユークリッドの公式の [1, 2]、ブラフマグプタの公式の [1, 3] に対応する」で済む話です。「いや、まだユークリッドの公式とブラフマグプタの公式が正しいことを納得できていない」という人もいそうですが、パソコンがあったら「全部試す」だけでわかります。で、実際にやったら確かに最小でした。既約ピタゴラス数の積は 60 の倍数である」は、e が 4 の倍数なのだから、「3 および 5 が o, e, d のうちいずれか一つにだけ入っている」という条件がつきます。でないと「原始」になりませんから。これも剰余系を使って力業で解けば簡単にいけますが、けっこう面倒くさいのが玉に瑕です。「直角二等辺三角形の辺長は既約ピタゴラス数ではない」は、「正方形の対角線は無理数である」と同じ話です。そうなると「背理法を使えばいい」という話になりそうですが、「背理法被害者の会」の理科大の安倍先生によれば「研究者になる前に背理法を覚えると脳が腐る」そうなので、背理法を使わずに、直接証明を狙いましょう。ただし、「素因数分解の一意性」を使わないと証明でかなり面倒臭くなるそうで、ガウスの証明まで追いかけてゆくとかなり大変です。最後は「o × e × d は六十の倍数である。」です。e が 4 の倍数なので、あとは o, e, d のいずれかに 3 と 5 が入っていることを示せばいいわけですが、これは剰余系に頼ることになります。
  mod 3 で考えると {0, 1, 2} なので自乗すると {0, 1, 1} となって {0, 1} となり、「いちおう射程内か？」という感じです。
  mod 5 だと {0, 1, 2, 3, 4} となり、自乗は {0, 1, 4, 4, 1} で {0, 1, 4} になります。「だけど、それって網羅したことになってるのか？」と考えると、これもまたけっこう面倒臭い話になります。

第 5 章：その他の三角形

ヘロン三角形

「ヘロンの公式」は、名前だけは覚えている人はいるでしょう。三角形の三辺の長さから面積 S と求める公式です。
三角形の三辺の長さを {A, B, C} とし、数 s を $\frac{A+B+C}{2}$ とします。すると面積は
$S = \sqrt{s \times (s - A) \times (s - B) \times (s - C)}$ であるという話です。実際に試してみましょう。
最小の既約ピタゴラス数は {3, 4, 5} です。
このとき s は $\frac{3+4+5}{2}$ なので、6 になります。
そうすると S は $\sqrt{s \times (s - A) \times (s - B) \times (s - C)} = \sqrt{6 \times (6 - 3) \times (6 - 4) \times (6 - 5)}$ となって 6 × 3 × 2 × 1 となって、素因数分解すると 2^2・3^2となって 6^2なので面積は 6、ということになります。底辺×高さ÷2 なので合ってます。<>　じゃあ、つぎは二等辺三角形でいってみましょう。底辺 6、高さ 4、等し

い二辺の長さは5です。ホントに12になるかな？
$s = (6 + 5 + 5) \div 2 = 8$
$S = 8 \times (8 - 6) \times (8 - 5) \times (8 - 5) = 8 \times 2 \times 3 \times 3 = 2^4 \times 3^2$ なので平方に開くと12。
　同じことを{8. 5, 5}でやったらどうなるかな？と考えると、果たして12（古い言い回しだなぁ）。つまり、ある既約ピタゴラス数がひとつあれば、「三辺の長さと面積が自然数の二等辺三角形が二つ得られる」ということになります。
「だったら二つの既約ピタゴラス三角形ではどうか？」「そもそも、偶数辺または奇数辺が同一の異なった既約ピタゴラス三角形は存在するのか？」「べつに『既約』にこだわらなくてもいいんじゃないか？」という発想も出てきます。そうなると「三辺の長さと面積が自然数の二等辺三角形」は無限個あるのだから、もうちょい範囲は広げられないか？　という話にもつながります。
　ここで気になあるのが、以下の図です。



　三角形ABCに内接する円を考えます。各辺が内接円に接する点までの長さをa, b, cとすると、各辺の長さは $\overline{\mathrm{AB}}$
が $a + b$、$\overline{\mathrm{AC}}$
が $a + c$、$\overline{\mathrm{BC}}$
が $b + c$ になります。そうなると、ここに直角三角形が六つあるということになります。ここで内接円の半径をrとおくと、$ra + rb + rc$ が三角形ABCの面積だということになります。要するにrをなんとか求めると三辺の長さから三角形ABCの面積がわかるということになります。
　じゃあ、rの長さはどれほどでしょうか？
　内接円の中心をPとします。ここから辺AB・AC・BCに垂線を降ろしたときの角度は直角です。そうすると三平方の定理が使えるわけで、 面積Sは（必ずしも既約でない）三つのバビロニア長方形の面積の和になります。こうなると、「三辺の長さと面積が自然数の不等辺三角形は存在するか？」という存在証明の話になります。これが見つかったとすると、「a, b, cにはどのような関係が成立するのか？」という話になります。
　プログラム言語Perlの開発者であるラリー・ウォールによれば、プログラマの三大美徳は「傲慢（Hubris）」「短気（Impatience）」「怠惰（Laziness）」だそうですから、ですから本物のプログラマは「俺なら解ける」「すぐにやってやる」「ただし面倒臭いことは全部コンピュータにやらせる」のが定跡です。　「はじめに」で述べたように、「あらゆる科学にとって大切な思考法」（のひとつ。これ意外にも批判的合理主義で知られるカール・ライムント・ポパー(1902 - 1994)がいう「アブダクション」があります）であるアプローチも

- 証明された法則の適用（Plan - 計画）
- 特殊の場合についての実験（Do - 実行）
- 一般法則の推測 (Check - 測定・評価）
- 法則の証明（Action - 対策・改善）

というPDCAサイクルと本質は同じではないか、ということになります。
　これが、なんかしら「$a \times b \times c$」を「$a + b + c$」で割った値が平方数だと三辺の長さと面積が自然数になりそうなんですよ。これ、2013年くらいには思いついていたのですが、プログラムを書く気力と体力がなくて放（ほ）ったらかしでした。そんなわけで昔のプログラムのコードを引っ張り出して数値実

験の準備をしています。

アイゼンシュタイン三角形

アイゼンシュタイン三角形は、三辺 {a, b, c} の間に $a^2 + ab + b^2 = c^2$ が成立する三角形です。アイゼンシュタイン三角形には「ひとつの角が 120°である」という特徴があります。最小のものは「名古屋{7, 5, 8}三角形」です。二等辺三角形に近いアイゼンシュタイン三角形からは三の平方根の近似値が得られます。似たようなものとしてはタレス（タレース）の三角形という「ひとつの角が 60°である」というものがあるのですが、これは正三角形との相性が悪い（無理数が出てくるので整三角形になりません）せいかあまり着目されません。正三角形以外で最小のものは「質屋さん{7, 8, 3}三角形」です。これは $2 \times 2^2／3^2$に落ちるので、このあたりに鍵がありそうに思います。ただ私は数学“愛好家”であって数学屋ではないので、「ヘンな女に引っかかっちゃったら」とボヤきつつ数式をいじくっています。

今後の展望

この方面にはいろいろ数値実験のネタがぞろぞろ出てきます。ここから派生して各種の整数論的な話題にも関連してきます。これを紙に印刷するとけっこう量があるし、紙の書籍だとプログラムを書くときに使いしづらいという欠点があります。

「暗黒通信団」という出版社からは、『π』という紙の書籍が出ていて、円周率が100,000,000桁まで（一億桁です。古い版は百万＝1,000,000桁でした）掲載されていて、212 ページ 690円です。これはマチンの式をつかっているようです。　素数を暗証番号にする人も多く、日付けとかぶらないように$1231 < p < 10000$ の範囲を網羅的に散策するとか、素数の原始根を最小のものだけではなくすべて求めるとか、Java で 214748347 以下の整数をすべて素因数分解できるロジックを書くとか、いろいろ楽しめます。

実用面では、ビットマップ画像の拡大・縮小があります。論文では画像と表がつきものですが、ブラウザ側の処理に時間がかかってタイムアウトしてしまうことが昔はありました（サーバー用のマシンには、普通はグラフィックボードが詰んでありません）。そこで画像と表を区別なく拡大・縮小するためのアルゴリズムを開発して操作性を向上させたこともあります。

このノリで話を続けてゆくとたいていの人は嫌気がさしてくる（プログラマの中にはハマる人はいそうというか、見事にハマったバカがここに一人いますが（笑））ので、このあたりで打ち止めにしておきます。

（了）

人名年表

- ピタゴラス(Pythagoras。BC582 - BC496）
- プラトン(BC427 - BC347)
- エウクレイデス(ユークリッド。紀元前三世紀?)
- ディオファントス(アレクサンドリアのディオファントス。三世紀初頭 - 三世紀末)
- ブラフマグプタ(598 – 665?)
- バースカラ二世(1114 - 1185）
- フィボナッチ（レオナルド・ダ・ピサ。「フィボナッチ」は「ボナッチの息子」の意。1170 - 1250）
- ピエール・ド・フェルマー(1607 - 1665)
- カール・フリードリヒ・ガウス(1777 - 1855)
- アイゼンシュタイン(アイゼンスタイン。1823 - 1852。師ガウスの四十歳以上年下なのにガウスより先に歿した夭折の天才)
- アンドレ・ヴァイユ(1906 - 1998)
-

用語索引

参考文献

- 吉田 洋一『零の発見 — 数学の生いたち』（岩波新書。1934）
- 遠山 啓『初等整数論』（岩波新書。1934）
- 一松 信『整数とあそぼう』(日本評論社。2006)

島田 正雄

　生まれは大田区大森の町工場街。

　日本大学航空宇宙工学科修了。高校生のころからの電算屋。「業務」と「システム」を繋ぐことに興味がある。

　とはいえ自分自身の興味のためにプログラムを書くのも好き。これも「仕事」という顧客さんと技術屋のあいだの空気の中で泳ぐための水練だと思っている。

　「人間とコンピュータの関係」とかいった大げさなことではなく、「役に立つパソコン」を追求しているうちに、日本語処理・数学・歴史（数学史・技術史）などに関心を持ち、教育について興味を持っている。

　現代人は「顔が見える個人」ではあるはずだが、古代バビロニアの書記の方々とは友人になれるのではないかと思っている。
