

# Meccano pentagons

<https://github.com/heptagons/meccano/penta>

## Abstract

We construct **two types** of meccano <sup>1</sup> regular pentagons. We use five equal strips to build the polygon perimeter and then we attach **internal diagonals** to make the polygon regular and rigid.

For each type we test **meccano rods** diagonals positions. All except one diagonal, the last one, have integer length always. When the iterations find the last diagonal is an integer too, we have a solution.

Testing using simple floating number calculations miss a lot of solutions due to the rounding errors accumulation, so we first find an algebraic equation which express the last diagonal in function of the rest of variables.

Several programs use the algebra formulas and conditions and iterate over a given range of increasing sizes to look for solutions preventing the repetitions by scaling.

From the two types of pentagons and the results obtained two conjectures emerges. **First conjecture** is that the first type of pentagon seems to have a **unique** solution after testing pentagons sides somehow large.

**Second conjecture** appears in second type of pentagon. For this type we got apparently infinite solutions but by the numeric value of the last diagonal called  $e$  seems to be always in the form  $10x + 1$  for  $x = 1, 2, 3, \dots$

## 1 Regular pentagon type 1

### 1.1 Type 1 algebra

Figure 1 show the layout of the meccano regular pentagon of type 1. Let define the side of the pentagon as  $a$  and define other three variables  $b$ ,  $c$  and  $d$ :

$$a = \overline{BC}, \quad b = \overline{BF}, \quad c = \overline{FI}, \quad d = \overline{CI}$$

By the figure the angles  $\angle LBC$  and  $\angle JFI$  are equal to  $2\pi/5$  so:

$$\alpha \equiv \frac{2\pi}{5} \tag{1}$$

$$\overline{BL} = a \cos \alpha \tag{2}$$

$$\overline{CL} = a \sin \alpha \tag{3}$$

$$\overline{FJ} = c \cos \alpha \tag{4}$$

$$\overline{IJ} = c \sin \alpha \tag{5}$$

---

<sup>1</sup> Meccano mathematics by 't Hooft

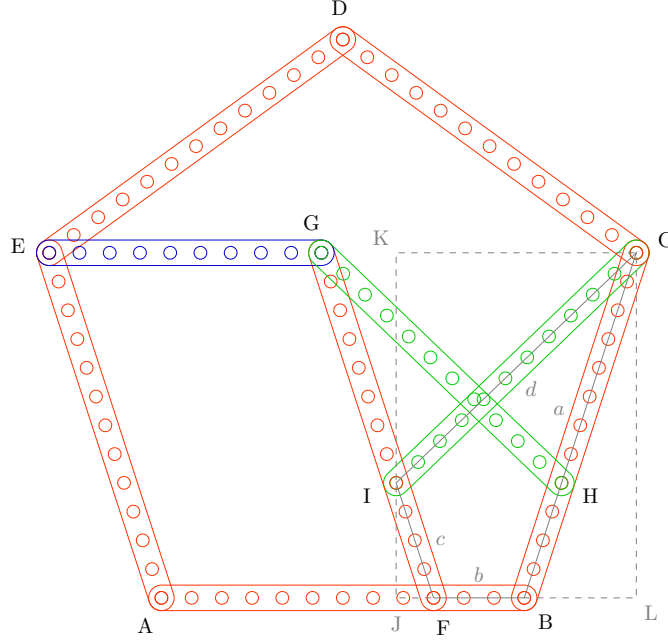


Figure 1: Pentagon of type 1.

For type 1 we have four variables and one angle. Let calculate  $d$  in function of  $a$ ,  $b$  and  $c$ :

$$\begin{aligned}
 d^2 &= (\overline{GI})^2 = (\overline{GK})^2 + (\overline{IK})^2 \\
 &= (\overline{BL} + \overline{BF} + \overline{FJ})^2 + (\overline{CL} - \overline{IJ})^2 \\
 &= (a \cos \alpha + b + c \cos \alpha)^2 + (a \sin \alpha - c \sin \alpha)^2 \\
 &= ((a + c) \cos \alpha + b)^2 + ((a - c) \sin \alpha)^2 \\
 &= (a + c)^2 \cos^2 \alpha + 2(a + c)b \cos \alpha + b^2 + (a - c)^2 \sin^2 \alpha \\
 &= (a^2 + c^2)(\cos^2 \alpha + \sin^2 \alpha) + 2ac(\cos^2 \alpha - \sin^2 \alpha) + 2(a + c)b \cos \alpha + b^2 \\
 &= (a^2 + c^2) + 2ac(\cos^2 \alpha - \sin^2 \alpha) + 2(a + c)b \cos \alpha + b^2
 \end{aligned} \tag{6}$$

For  $\alpha = 2\pi/5$  we will use the following common pentagon identities:

$$\cos \alpha = \frac{-1 + \sqrt{5}}{4} \tag{8}$$

$$\cos^2 \alpha = \frac{3 - \sqrt{5}}{8} \tag{9}$$

$$\sin^2 \alpha = \frac{5 + \sqrt{5}}{8} \tag{10}$$

$$\cos^2 \alpha - \sin^2 \alpha = -\frac{1 + \sqrt{5}}{4} \tag{11}$$

Applying the identities to the last equation of  $d$  we get:

$$\begin{aligned}
d^2 &= a^2 + c^2 - \left(\frac{1+\sqrt{5}}{2}\right)ac + \left(\frac{-1+\sqrt{5}}{2}\right)(a+c)b + b^2 \\
&= a^2 + c^2 - \frac{ac}{2} - \frac{(a+c)b}{2} + b^2 + \left(-\frac{ac}{2} + \frac{(a+c)b}{2}\right)\sqrt{5} \\
&= a^2 + b^2 + c^2 - \frac{ac + (a+c)b}{2} + \left(\frac{-ac + (a+c)b}{2}\right)\sqrt{5}
\end{aligned} \tag{12}$$

Let define two variables  $p$  and  $q$  such that  $d^2 = p + q\sqrt{5}$  so we have:

$$\begin{aligned}
d^2 &= p + q\sqrt{5} \\
q &= \frac{-ac + (a+c)b}{2} \\
p &= a^2 + b^2 + c^2 - \frac{ac + (a+c)b}{2} \\
&= a^2 + b^2 + c^2 - \frac{-ac + (a+c)b}{2} - ac \\
&= a^2 + b^2 + c^2 - q - ac
\end{aligned}$$

For a meccano pentagon we need  $d$  to be an integer. If we let the integer  $q > 0$  then  $d = \sqrt{p + q\sqrt{5}}$  will never be an integer for  $p$  and  $q$  integers. If we force  $q$  to be zero then  $d = \sqrt{p}$  and  $d$  will have possibilities to be an integer.

So before calculating  $d$ , we **need** to force the condition  $q = 0$  which is equivalent to make  $-ac + (a+c)b = 0$ :

$$a > b, \quad a > c \tag{13}$$

$$ac = (a+c)b \tag{14}$$

$$d = \sqrt{a^2 + b^2 + c^2 + ac} \tag{15}$$

## 1.2 Type 1 program

First we write a **go** struct called **Sols** (see section 3) to store and print solutions eventually found. The function **Add** prevents duplicated solutions by scaling, comparing a prospect with the already collected:

The following function called **pentagons\_type\_1** iterate over three variables  $a \leq max$ ,  $1 \leq b \leq a$ ,  $0 \leq c \leq a$  (lines 15, 16 and 17). The  $q = 0$  condition mentioned above, is tested (in line 18) and only when the condition holds we check whether  $d$  is an integer (internal function called **check** at line 5). When  $d$  is an integer we call function **sols.Add** (line 11) to print and store the solution.

```

1 func pentagons_type_1(max int) {
2
3     sols := &Sols{}
4
5     check := func(a, b, c int) {
6         f := float64(a*a + b*b + c*c - a*c)
7         if f < 0 {
8             return
9         }
10        if d := int(math.Sqrt(f)); math.Pow(float64(d), 2) == f {
11            sols.Add(a, b, c, d)
12        }
13    }

```

```

14
15   for a := 1; a < max; a++ {
16       for b := 1; b <= a; b++ {
17           for c := 0; c <= a; c++ {
18               if a*c == (a + c)*b {
19                   check(a, b, c)
20               }
21           }
22       }
23   }
24 }

```

### 1.3 Type 1 results

After serching for values of  $a \leq 5000$  we found a single result:

```

1   1   a= 12 b=   3 c=   4 d= 11

```

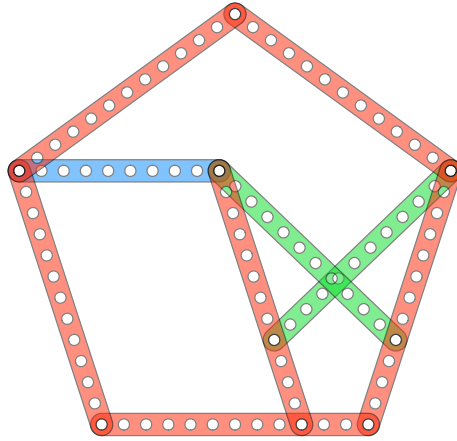


Figure 2: The smallest and maybe unique pentagons of type 1. Is composed of 6 rods of length  $a = 12$  in color red, two rods of length  $d = 11$  in green and one strip of size  $a - b = 9$  in blue.

Figure 2 shows the first (unique?) pentagon of type 1 with values  $a = 12$ ,  $b = 3$ ,  $c = 4$  and  $d = 11$ .

### 1.4 Type 1 conjecture

There is only a single case for the type 1 with values  $a = 12$ ,  $b = 3$ ,  $c = 4$  and  $d = 11$ .

## 2 Regular pentagon type 2

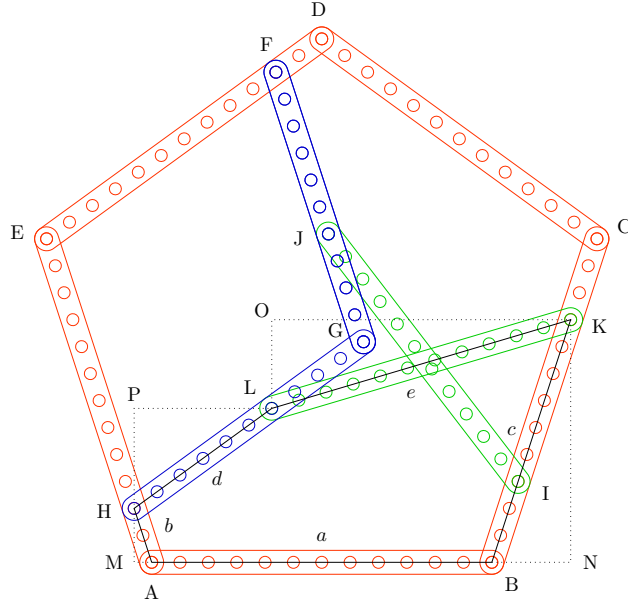


Figure 3: Pentagon of type 2.

### 2.1 Type 2 algebra

Figure 3 show the layout of the meccano regular pentagon of type 2. Let define the side of the pentagon as  $a$  and define other four variables  $b$ ,  $c$ ,  $d$  and  $e$ :

$$a = \overline{AB}, \quad b = \overline{AH}, \quad c = \overline{BK}, \quad d = \overline{HL}, \quad e = \overline{KL}$$

From the figure we note the  $\angle NBC = 2\pi/5$  and  $\angle MAH = \pi/5$  so we have:

$$\alpha = \frac{2\pi}{5}, \quad \beta = \frac{\pi}{5} \tag{16}$$

$$\overline{BN} = b \cos \alpha, \quad \overline{KN} = b \sin \alpha \tag{17}$$

$$\overline{AM} = c \cos \alpha, \quad \overline{HM} = c \sin \alpha \tag{18}$$

$$\overline{LP} = d \cos \beta, \quad \overline{HP} = d \sin \beta \tag{19}$$

We express  $e$  as a function of variables  $a$ ,  $b$ ,  $c$  and  $d$ .  $e^2$  equals  $(\overline{KO})^2 + (\overline{LO})^2$  so we first calculate  $\overline{KO}$  and  $\overline{LO}$ . From figure 3:

$$\begin{aligned} \overline{KO} &= \overline{AM} + \overline{AB} + \overline{BN} - \overline{LP} \\ &= b \cos \alpha + a + c \cos \alpha - d \cos \beta \\ &= (b + c) \cos \alpha + a - d \cos \beta \end{aligned} \tag{20}$$

$$\begin{aligned} \overline{LO} &= \overline{KN} - \overline{HM} - \overline{HP} \\ &= c \sin \alpha - b \sin \alpha - d \sin \beta \\ &= (c - b) \sin \alpha - d \sin \beta \end{aligned} \tag{21}$$

So by adding the squares we get:

$$\begin{aligned}
e^2 &= (\overline{KO})^2 + (\overline{LO})^2 \\
&= ((b+c)\cos\alpha)^2 + 2(b+c)\cos\alpha(a-d\cos\beta) + (a-d\cos\beta)^2 \\
&\quad + ((c-b)\sin\alpha)^2 - 2(c-b)\sin\alpha d\sin\beta + (d\sin\beta)^2 \\
&= (b^2+c^2)(\cos^2\alpha + \sin^2\alpha) + 2bc(\cos^2\alpha - \sin^2\alpha) \\
&\quad + 2a(b+c)\cos\alpha - 2(b+c)d\cos\alpha\cos\beta - 2(c-b)d\sin\alpha\sin\beta \\
&\quad + a^2 - 2ad\cos\beta + d^2(\cos^2\beta + \sin^2\beta) \\
&= a^2 + b^2 + c^2 + d^2 + 2bc\cos(2\alpha) \\
&\quad + 2a(b+c)\cos\alpha - 2(b+c)d\cos\alpha\cos\beta - 2(c-b)d\sin\alpha\sin\beta - 2ad\cos\beta \\
&= a^2 + b^2 + c^2 + d^2 + 2bc\cos(2\alpha) + 2a(b+c)\cos\alpha \\
&\quad - 2((b+c)\cos\alpha\cos\beta + (c-b)\sin\alpha\sin\beta + a\cos\beta)d \\
&= \boxed{a^2 + b^2 + c^2 + d^2 + 2bc\cos(2\alpha) + 2a(b+c)\cos\alpha - 2(b\cos(\alpha+\beta) + c\cos(\alpha-\beta) + a\cos\beta)d} \quad (22)
\end{aligned}$$

We will use the following identities for angles  $\alpha = 2\pi/5$  and  $\beta = \pi/5$ :

$$2\cos\alpha = \frac{-1+\sqrt{5}}{2} \quad (23)$$

$$2\cos\beta = \frac{1+\sqrt{5}}{2} = 2\cos(\alpha-\beta) \quad (24)$$

$$2\cos(2\alpha) = -2\cos\beta \quad (25)$$

$$2\cos(\alpha+\beta) = -2\cos\alpha \quad (26)$$

We replace identities in  $e^2$  equation:

$$\begin{aligned}
e^2 &= a^2 + b^2 + c^2 + d^2 + bc\left(\frac{-1-\sqrt{5}}{2}\right) + a(b+c)\left(\frac{-1+\sqrt{5}}{2}\right) \\
&\quad - bd\left(\frac{1-\sqrt{5}}{2}\right) - cd\left(\frac{1+\sqrt{5}}{2}\right) - ad\left(\frac{1+\sqrt{5}}{2}\right) \\
&= a^2 + b^2 + c^2 + d^2 - \frac{bc+ab+ac+bd+cd+ad}{2} + \frac{-bc+ab+ac+bd-cd-ad}{2}\sqrt{5} \\
&= a^2 + b^2 + c^2 + d^2 - \frac{(a+b)(c+d)+ab+cd}{2} + \frac{(a-b)(c-d)+ab-cd}{2}\sqrt{5} \quad (27)
\end{aligned}$$

Let define two variables  $p$  and  $q$  such that  $e^2 = p + q\sqrt{5}$ :

$$\begin{aligned}
p &\equiv a^2 + b^2 + c^2 + d^2 - \frac{(a+b)(c+d)+ab+cd}{2} \\
q &\equiv \frac{(a-b)(c-d)+ab-cd}{2}
\end{aligned}$$

For a meccano pentagon we need  $e$  to be an integer. If we let the integer  $q > 0$  then  $e = \sqrt{p+q\sqrt{5}}$  will never be an integer for  $p$  and  $q$  integers. If we force  $q$  to be zero then  $e = \sqrt{p}$  has possibilities to be an integer. So before calculating  $e$  we **need** to force the condition that  $q = 0$  or that is the same  $cd = (a-b)(c-d) + ab$ :

$$a > b, \quad a > c \quad (28)$$

$$cd = (a-b)(c-d) + ab \quad (29)$$

We can use this  $cd$  value to simplify  $p$ :

$$\begin{aligned}
p &= a^2 + b^2 + c^2 + d^2 - \frac{(a+b)(c+d) + ab + cd}{2} \\
&= a^2 + b^2 + c^2 + d^2 - \frac{(a+b)(c+d) + ab + (a-b)(c-d) + ab}{2} \\
&= a^2 + b^2 + c^2 + d^2 - ac - bd - ab
\end{aligned}$$

So finally, when  $q = 0$  we calculate  $e = \sqrt{p}$  expecting to be an integer:

$$e = \sqrt{a^2 + b^2 + c^2 + d^2 - ac - bd - ab} \quad (30)$$

Another solution is:

$$e = \sqrt{a^2 + b^2 + c^2 + d^2 - ad - bc - cd} \quad (31)$$

## 2.2 Type 2 first program

With the type 2 equations ready, we use the next function to search the solutions. Is called `pentagons_type_2`, iterates over the integer values of rods  $a$  (line 15),  $b$  (line 16),  $c$  (line 17) and  $d$  (line 18) to discover a strip  $e$  with integer length too. First we check condition  $q == 0$  is true (line 19) and square root is integer (line 10):

```

1 func pentagons_type_2(max int) {
2
3     sols := &Sols{}
4
5     check := func(a, b, c, d int) {
6         f := float64(a*a + b*b + c*c + d*d - a*c - b*d - a*b)
7         if f < 0 {
8             return
9         }
10        if e := int(math.Sqrt(f)); math.Pow(float64(e), 2) == f {
11            sols.Add(a, b, c, d, e)
12        }
13    }
14
15    for a := 1 ; a < max; a++ {
16        for b := 1; b < a; b++ {
17            for c := 0; c < a; c++ {
18                for d := 1; d < a; d++ {
19                    if ((a - b)*(c - d) + a*b == c*d) {
20                        check(a, b, c, d)
21                    }
22                }
23            }
24        }
25    }
26 }

```

## 2.3 Type 2 first results

The program found 19 pentagons of type 2 for  $a \leq 100$ . While we found a single solution for type 1, type 2 has several.

```

1  1  a= 12 b=  2 c=  9 d=  6 e= 11
2  2  a= 12 b=  3 c=  0 d=  4 e= 11
3  3  a= 12 b=  6 c=  3 d= 10 e= 11
4  4  a= 31 b=  4 c= 28 d= 16 e= 31
5  5  a= 31 b= 15 c=  3 d= 27 e= 31
6  6  a= 38 b= 12 c= 18 d= 21 e= 31
7  7  a= 38 b= 17 c= 20 d= 26 e= 31
8  8  a= 48 b=  8 c= 24 d= 21 e= 41
9  9  a= 48 b= 12 c=  9 d= 20 e= 41
10 10  a= 48 b= 27 c= 24 d= 40 e= 41
11 11  a= 48 b= 28 c= 39 d= 36 e= 41
12 12  a= 72 b= 21 c= 48 d= 40 e= 61
13 13  a= 72 b= 24 c= 16 d= 39 e= 61
14 14  a= 72 b= 32 c= 24 d= 51 e= 61
15 15  a= 72 b= 33 c= 56 d= 48 e= 61
16 16  a= 78 b= 27 c=  4 d= 42 e= 71
17 17  a= 78 b= 36 c= 74 d= 51 e= 71
18 18  a= 87 b= 28 c= 36 d= 48 e= 71
19 19  a= 87 b= 39 c= 51 d= 59 e= 71

```

## 2.4 Type 2 simpler program

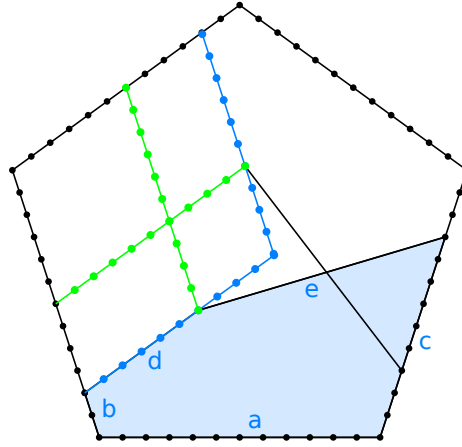


Figure 4: Pentagon of type 2 has a symmetry where pair bars green and blue can be switched leaving the strip  $e$  lengths and positions unmodified. This symmetry appears in the first program when two solutions have same  $a$  and same  $e$ .

Figure 4 show what happens when the first program reports two solutions with the same  $a$  and the same  $e$ . The type 2 symmetry can be taken into account to simplify the first program to reduce the search space and report only the half of symmetries. Next go function called `pentagons_type_2_half` first iterates over  $1 \leq a \leq \max$  (line 4), then over  $1 \leq b < a$  (line 6), then over  $1 \leq d < (a - b)$  (line 8) and finally over  $0 \leq c < a$  (line 10).

```

1  func pentagons_type_2_half(max int) {
2      sols := &Sols{}
3      aa, a_b, ab, bb, dd, ad, bc, c_d, cd, cc := 0,0,0,0,0,0,0,0,0,0
4      for a := 1; a <= max; a++ {

```



```

5   aa = a*a
6   for b := 1; b < a; b++ {
7       a_b, ab, bb = a - b, a*b, b*b
8       for d := 1; d < (a-b); d++ {
9           dd, ad = d*d, a*d
10          for c := 0; c < a; c++ {
11              bc, c_d, cd, cc = b*c, c - d, c*d, c*c
12              if a_b * c_d + ab == cd {
13                  if f := float64(aa + bb + cc + dd - ad - bc - cd); f > 0 {
14                      if e := int(math.Sqrt(f)); math.Pow(float64(e), 2) == f {
15                          sols.Add(a, b, c, d, e)
16                      }
17                  }
18              }
19          }
20      }
21  }
22 }
23 }

```

## 2.5 Type 2 simpler results

The second type 2 program found 139 solutions iterating over  $1 \leq a \leq 1000$ :

1	1	a= 12	b= 2	c= 9	d= 6	e= 11
2	2	a= 12	b= 3	c= 0	d= 4	e= 11
3	3	a= 31	b= 4	c= 28	d= 16	e= 31
4	4	a= 38	b= 12	c= 18	d= 21	e= 31
5	5	a= 48	b= 8	c= 24	d= 21	e= 41
6	6	a= 48	b= 12	c= 9	d= 20	e= 41
7	7	a= 72	b= 21	c= 48	d= 40	e= 61
8	8	a= 72	b= 24	c= 16	d= 39	e= 61
9	9	a= 78	b= 27	c= 4	d= 42	e= 71
10	10	a= 87	b= 28	c= 36	d= 48	e= 71
11	11	a=111	b= 39	c= 99	d= 67	e=101
12	12	a=121	b= 33	c= 33	d= 57	e=101
13	13	a=128	b= 8	c= 89	d= 56	e=121
14	14	a=138	b= 12	c= 54	d= 47	e=121
15	15	a=145	b= 45	c= 39	d= 75	e=121
16	16	a=147	b= 43	c= 51	d= 75	e=121
17	17	a=151	b= 19	c= 73	d= 61	e=131
18	18	a=156	b= 43	c= 96	d= 84	e=131
19	19	a=165	b= 36	c=132	d= 88	e=151
20	20	a=179	b= 15	c=177	d= 93	e=191
21	21	a=183	b= 66	c= 62	d=108	e=151
22	22	a=201	b= 9	c= 13	d= 21	e=191
23	23	a=204	b= 21	c=112	d= 84	e=181
24	24	a=216	b= 48	c=111	d=104	e=181
25	25	a=236	b= 80	c= 20	d=125	e=211
26	26	a=249	b= 45	c= 75	d= 95	e=211
27	27	a=264	b= 76	c= 3	d=108	e=241
28	28	a=285	b= 73	c= 27	d=111	e=251
29	29	a=296	b=104	c=128	d=173	e=241
30	30	a=303	b= 51	c= 29	d= 81	e=271
31	31	a=304	b= 76	c=133	d=148	e=251
32	32	a=312	b= 36	c= 93	d=100	e=271
33	33	a=315	b= 24	c=160	d=120	e=281
34	34	a=324	b= 64	c=204	d=159	e=281
35	35	a=343	b= 7	c=115	d= 91	e=311
36	36	a=352	b= 3	c=240	d=144	e=341
37	37	a=354	b= 53	c= 60	d=102	e=311
38	38	a=368	b= 36	c=219	d=156	e=331
39	39	a=369	b= 37	c= 27	d= 63	e=341
40	40	a=370	b= 1	c=172	d=118	e=341
41	41	a=375	b= 15	c=191	d=135	e=341
42	42	a=378	b= 21	c= 84	d= 86	e=341
43	43	a=384	b=120	c=312	d=223	e=341
44	44	a=390	b= 84	c= 50	d=135	e=341
45	45	a=390	b= 87	c=228	d=194	e=331
46	46	a=392	b=119	c=296	d=224	e=341
47	47	a=392	b=128	c= 56	d=203	e=341
48	48	a=393	b= 98	c= 54	d=156	e=341
49	49	a=396	b=138	c= 73	d=222	e=341
50	50	a=399	b= 70	c=210	d=180	e=341
51	51	a=403	b= 78	c=114	d=156	e=341
52	52	a=404	b= 89	c=104	d=164	e=341
53	53	a=408	b= 16	c=312	d=183	e=401
54	54	a=408	b= 84	c=167	d=180	e=341
55	55	a=411	b=123	c=243	d=227	e=341
56	56	a=435	b= 96	c=400	d=240	e=421
57	57	a=450	b= 92	c=438	d=249	e=451
58	58	a=468	b=173	c= 24	d=276	e=431
59	59	a=480	b= 80	c= 75	d=144	e=421
60	60	a=486	b=180	c= 18	d=287	e=451
61	61	a=488	b= 72	c= 15	d= 96	e=451
62	62	a=488	b=132	c=423	d=276	e=451
63	63	a=488	b=152	c=269	d=272	e=401

64	64	a=495	b=135	c=415	d=279	e=451
65	65	a=502	b= 93	c= 36	d=138	e=451
66	66	a=507	b= 18	c=366	d=220	e=491
67	67	a=507	b= 60	c= 84	d=128	e=451
68	68	a=509	b=150	c= 42	d=228	e=451
69	69	a=516	b=114	c=169	d=222	e=431
70	70	a=520	b= 36	c=225	d=180	e=461
71	71	a=525	b=185	c=399	d=315	e=451
72	72	a=525	b=189	c=105	d=305	e=451
73	73	a=528	b= 80	c=171	d=192	e=451
74	74	a=540	b=150	c=321	d=290	e=451
75	75	a=543	b=123	c=221	d=249	e=451
76	76	a=546	b=135	c=228	d=262	e=451
77	77	a=552	b=179	c=288	d=312	e=451
78	78	a=553	b=180	c=276	d=312	e=451
79	79	a=560	b=200	c=344	d=335	e=461
80	80	a=565	b= 69	c=153	d=177	e=491
81	81	a=588	b=104	c= 12	d=135	e=541
82	82	a=600	b= 65	c=240	d=216	e=521
83	83	a=600	b=120	c= 96	d=205	e=521
84	84	a=617	b= 89	c=533	d=317	e=601
85	85	a=632	b=113	c=152	d=224	e=541
86	86	a=652	b= 58	c=235	d=214	e=571
87	87	a=661	b=109	c= 37	d=157	e=601
88	88	a=684	b=237	c=192	d=388	e=571
89	89	a=699	b= 84	c=564	d=344	e=671
90	90	a=701	b=254	c=698	d=428	e=671
91	91	a=713	b=234	c=582	d=420	e=631
92	92	a=715	b=211	c=655	d=415	e=671
93	93	a=720	b=216	c=712	d=423	e=701
94	94	a=724	b=147	c= 72	d=228	e=641
95	95	a=728	b= 21	c=192	d=168	e=661
96	96	a=729	b= 36	c=428	d=288	e=671
97	97	a=732	b= 18	c=681	d=358	e=781
98	98	a=732	b= 42	c=111	d=134	e=671
99	99	a=744	b=228	c=155	d=372	e=631
100	100	a=746	b=164	c= 38	d=233	e=671
101	101	a=755	b=123	c=267	d=291	e=641
102	102	a=756	b= 69	c=168	d=196	e=671

103	103	a=762	b= 73	c=372	d=294	e=671
104	104	a=765	b= 30	c=354	d=260	e=691
105	105	a=777	b=234	c=118	d=372	e=671
106	106	a=781	b=108	c=348	d=312	e=671
107	107	a=784	b=192	c=189	d=336	e=661
108	108	a=800	b=164	c=263	d=332	e=671
109	109	a=804	b=177	c=272	d=348	e=671
110	110	a=805	b=202	c=238	d=364	e=671
111	111	a=810	b=276	c=510	d=475	e=671
112	112	a=819	b=136	c=216	d=288	e=701
113	113	a=824	b=276	c=363	d=468	e=671
114	114	a=826	b=315	c=420	d=510	e=671
115	115	a=840	b=196	c=777	d=468	e=811
116	116	a=845	b=285	c=465	d=489	e=691
117	117	a=859	b=130	c=502	d=388	e=751
118	118	a=861	b=126	c= 66	d=196	e=781
119	119	a=863	b=303	c=711	d=519	e=761
120	120	a=864	b= 24	c=349	d=264	e=781
121	121	a=873	b=137	c=453	d=381	e=751
122	122	a=879	b=231	c= 63	d=343	e=781
123	123	a=885	b=206	c=642	d=468	e=781
124	124	a=885	b=309	c= 13	d=477	e=821
125	125	a=892	b=112	c=196	d=259	e=781
126	126	a=896	b=144	c=528	d=411	e=781
127	127	a=896	b=332	c=725	d=548	e=781
128	128	a=904	b=328	c=640	d=547	e=761
129	129	a=905	b=161	c=185	d=305	e=781
130	130	a=912	b=168	c=507	d=424	e=781
131	131	a=915	b=135	c=345	d=349	e=781
132	132	a=928	b=319	c=232	d=520	e=781
133	133	a=938	b=252	c=270	d=441	e=781
134	134	a=947	b=306	c=558	d=540	e=781
135	135	a=948	b=342	c=589	d=570	e=781
136	136	a=949	b=273	c=495	d=507	e=781
137	137	a=960	b=195	c=760	d=504	e=881
138	138	a=961	b=249	c=633	d=513	e=821
139	139	a=987	b=350	c=594	d=588	e=811

## 2.6 Type 2 conjecture

The last report of 139 pentagons shows all  $e$  values have the form  $10x + 1$  for  $x$  integer. So the conjecture is that  $e$  always is of the form  $10x + 1$  for  $x$  integer.

Next go function called `pentagons_type_2_half_with_conjecture` is an adaptation of the previous one and instead checking for a square root to be an integer, only checks for  $e^2 = (10x + 1)^2$  for small  $x > 1$ . The results of this program is exactly the same result of the program checking the square root, up to  $a \leq 1000$ .

```

1 func pentagons_type_2_half_with_conjecture(max int) {
2   sols := &Sols{}
3   aa, a_b, ab, bb, dd, ad, bc, c_d, cd, cc := 0,0,0,0,0,0,0,0,0,0
4   for a := 1; a <= max; a++ {
5     aa = a*a
6     for b := 1; b < a; b++ {
7       a_b, ab, bb = a - b, a*b, b*b
8       for d := 1; d < (a-b); d++ {
9         dd, ad = d*d, a*d

```

```

10     for c := 1; c < a; c++ {
11         bc, c_d, cd, cc = b*c, c - d, c*d, c*c
12         if a_b * c_d + ab == cd {
13
14             e2 := aa + bb + cc + dd - ad - bc - cd
15
16             x := 1
17             for {
18                 if e := 10*x + 1; e*e == e2 {
19                     sols.Add(a, b, c, d, e)
20                     break
21                 } else if e*e > e2 {
22                     break
23                 }
24                 x++
25             }
26         }
27     }
28 }
29 }
30 }
31 }

```

## 2.7 Type 2 examples

Figures 5, 6 and 7 show the first 18 pentagons of type 2 found.

## 3 Solutions code

Next code is where the solutions are compared against previous ones and eventually stored and printed. This code is used by pentagons and another polygons searches as mentioned in their respective articles.

```

1 type Sols struct {
2     sols [][]int
3 }
4
5 func (s *Sols) Add(rods ...int) {
6     if len(rods) < 0 {
7         return
8     }
9     const RODS = "abcdefghijkl"
10    for _, s := range s.sols {
11        a := rods[0]
12        if a % s[0] != 0 {
13            continue
14        }
15        // new a is a factor of previous a
16        f := a / s[0]
17        cont := false
18        for r := 1; r < len(rods); r++ {
19            if s[r] == 0 {
20                continue
21            }
22            b := rods[r]
23            if t := b % s[r] == 0 && b / s[r] == f; !t {

```

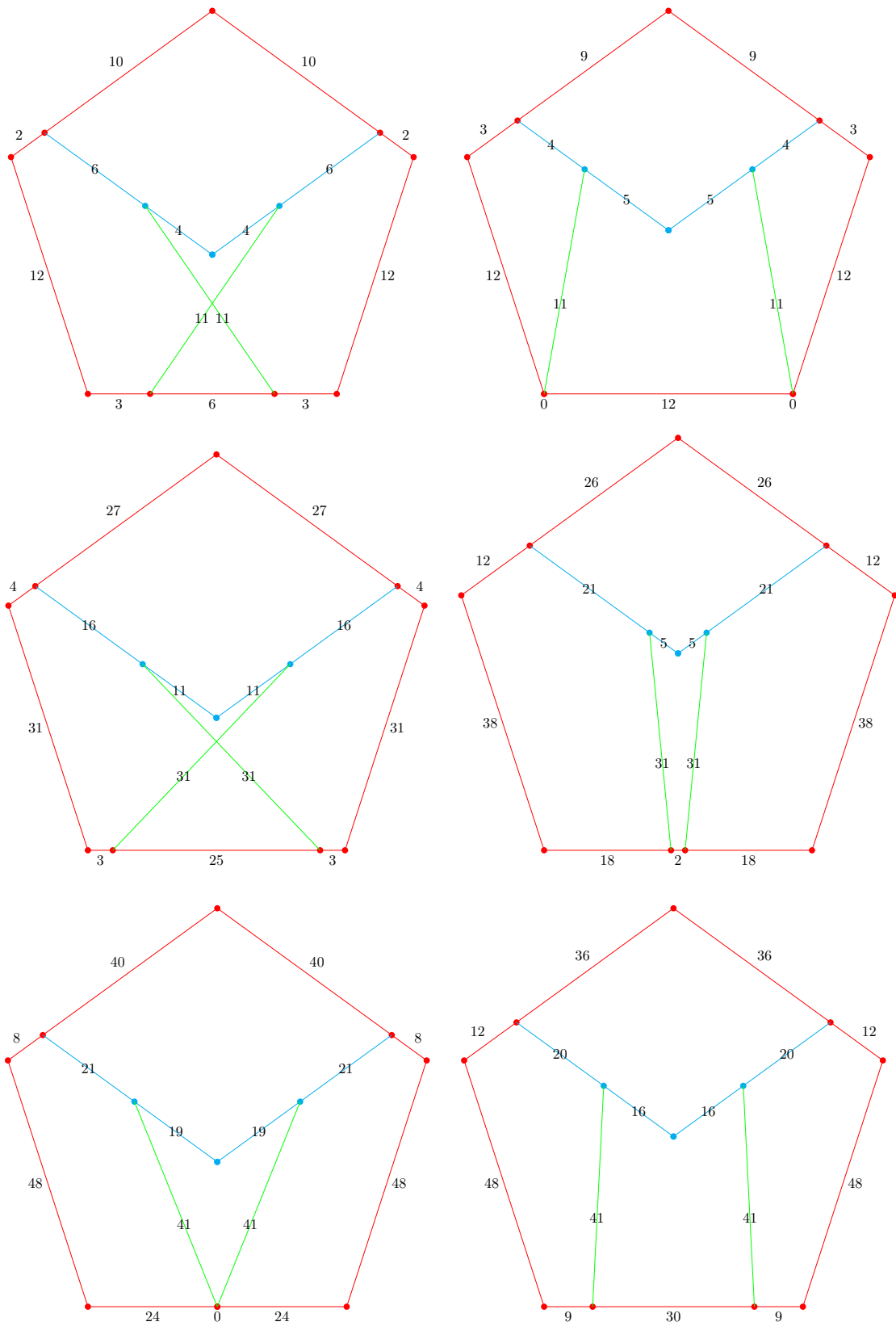


Figure 5: Pentagons 1-6 of type 2 with diagonals 11, 31 and 41.

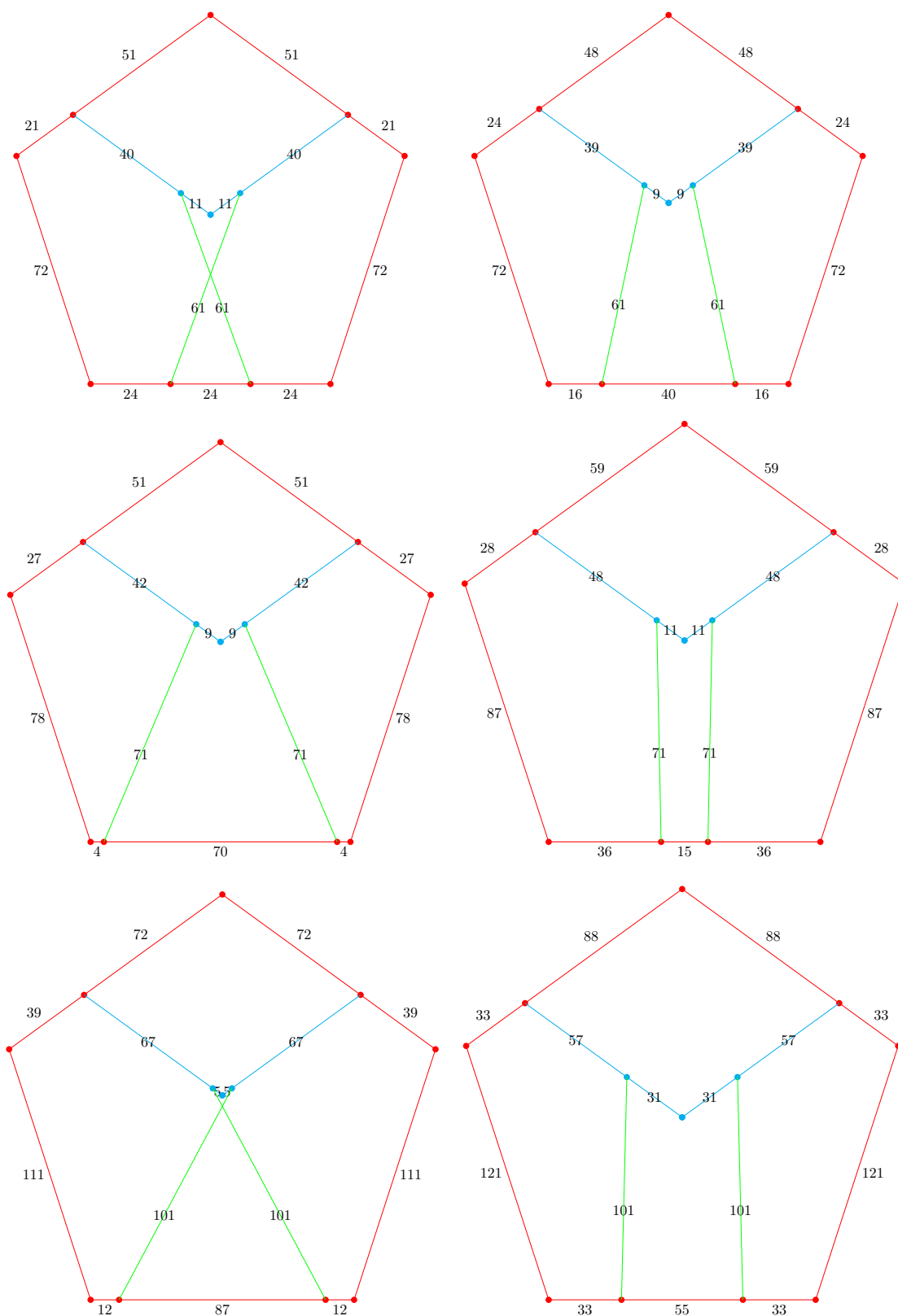


Figure 6: Pentagons 7-12 of type 2 with diagonals 61, 71 and 101.

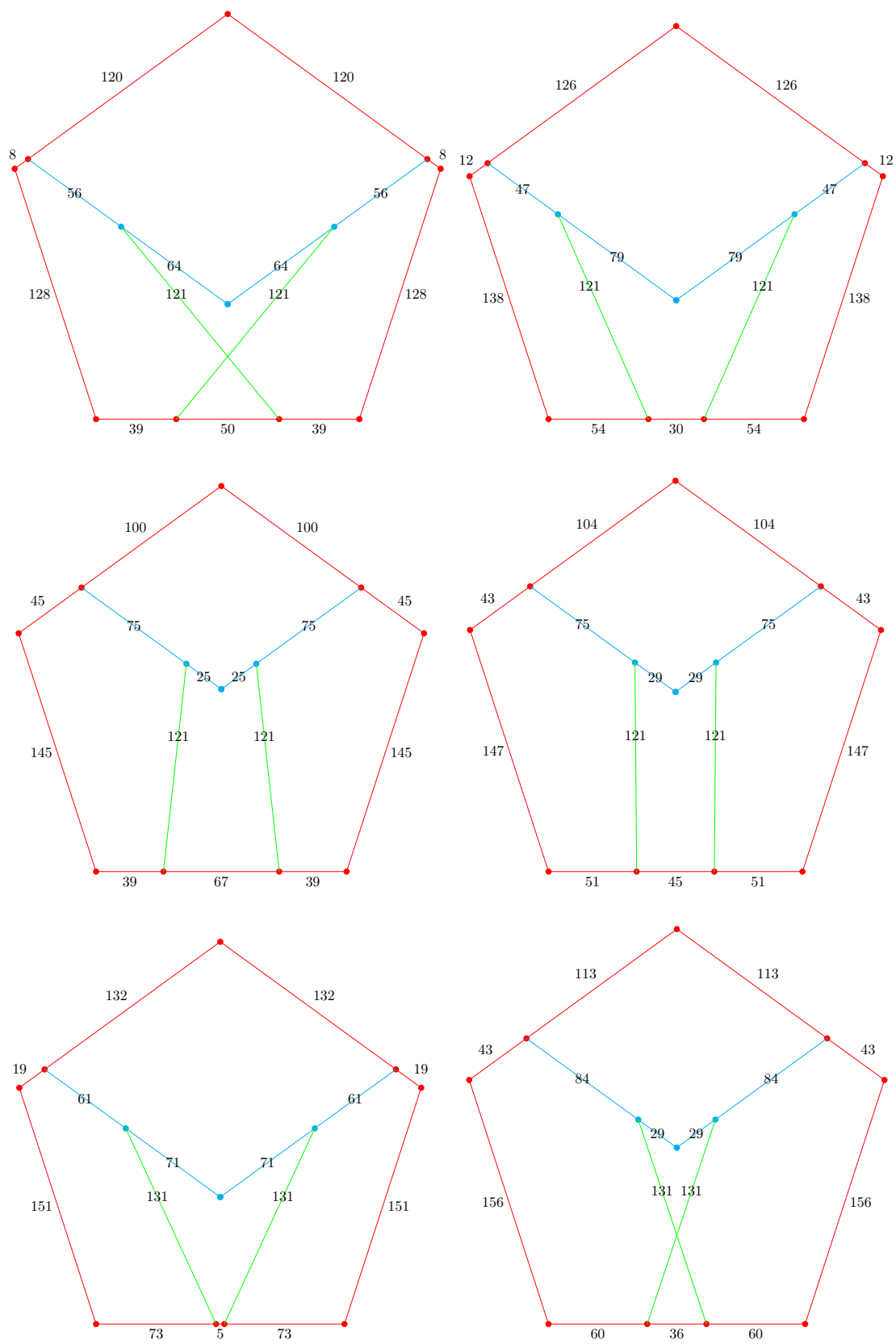


Figure 7: Pentagons 13-18 of type 2 with diagonals 121 and 131.

```

24         cont = true
25         break
26     }
27 }
28 if cont {
29     continue // scaled solution already found (reject)
30 }
31 return
32 }
33 // solution!
34 if s.sols == nil {
35     s.sols = make([][]int, 0)
36 }
37 s.sols = append(s.sols, rods)
38 fmt.Printf("%3d ", len(s.sols))
39 for i, r := range rods {
40     fmt.Printf(" %c=%3d", RODS[i], r)
41 }
42 fmt.Println()
43 }

```