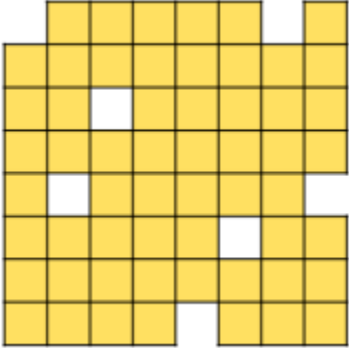



Enigmatic Code

Programming Enigma Puzzles

Puzzle #29: How many strips?
From New Scientist #3255, 9th November 2019 [\[link\]](#) [\[link\]](#)





How many 3×1 strips, like the example pictured [right], can be cut out of this piece of wood [left]?

[puzzle#29]

Rate this:

1 Rating

Share this:



Like

Be the first to like this.

Puzzle #31: Three hats
In "puzzle#"

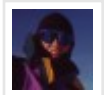
Enigma 217: Auntie
Greta's age
In "enigma"

Puzzle #24: Three stamps
In "puzzle#"

puzzle# by: David Bodycombe

[← Enigma 985: 57 varieties](#)[Enigma 525: Crossed lines →](#)**2 responses to “Puzzle #29: How many strips?”****Jim Randell** 9 November 2019 at 8:08 am

I assumed the puzzle was asking what is the maximum number of strips that can be cut of the piece of wood.



There are 7 squares missing from a complete 8×8 grid, so we cannot make more than 19 strips consisting of 3 squares from the piece of wood.

This Python program searches for the maximum possible number of strips. A minimal solution is found immediately, but the program takes 2m13s to run to completion.

```

1  from enigma import Accumulator, div, printf
2
3  # make the grid (0 can be used, -1 can't be used)
4  grid = [0] * 64
5  for i in [0, 6, 18, 33, 39, 45, 60]: grid[i] = -1
6
7  # try to tile the grid
8  # m = accumulator for count of unused squares
9  # g = current state of the grid
10 # u = number of unused squares
11 # i = current square to try
12 # v = next value to fill out on tile
13 def tile(m, g, u, i=0, v=1):
14     # are we done?
15     if i > 63:
16         if u < m.value: printf("[u={u} -> g={g}]")
17         m.accumulate(u)
18         return
19     elif g[:i].count(0) < m.value:
20         if g[i] == 0:
21             # can we place a vertical strip?
22             if i < 48 and 0 == g[i + 8] == g[i + 16]:
23                 g[i] = g[i + 8] = g[i + 16] = v
24                 tile(m, g, u - 3, i + 1, v + 1)
25                 g[i] = g[i + 8] = g[i + 16] = 0
26             # can we place a horizontal strip?
27             if (i % 8) < 6 and 0 == g[i + 1] == g[i + 2]:
28                 g[i] = g[i + 1] = g[i + 2] = v
29                 tile(m, g, u - 3, i + 3, v + 1)
30                 g[i] = g[i + 1] = g[i + 2] = 0
31             # move on to the next square
32             tile(m, g, u, i + 1, v)
33
34 # accumulate solutions by the smallest number of unused squares
35 u = grid.count(0)
36 m = Accumulator(fn=min, value=u)
37

```

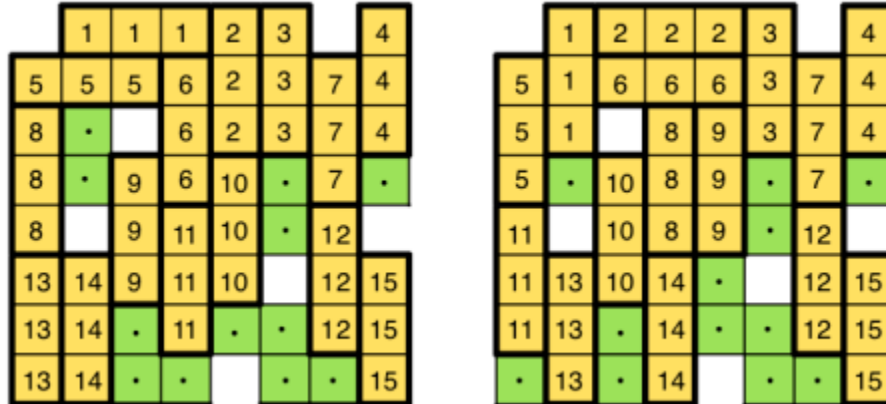
```

38 | # find tilings on the grid
39 | tile(m, grid, u)
40 |
41 | # output max number number of strips
42 | n = div(u - m.value, 3)
43 | printf("max = {n} strips")

```

Solution: 15 strips can be cut from the shape.

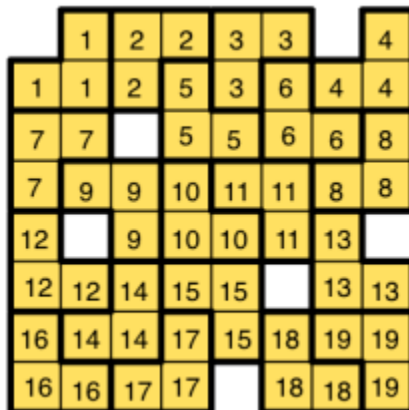
There are many ways to cut the 15 strips. Here are two:



The unused squares are marked in green.

These can be formed using a “greedy” strategy by starting in the top left corner and moving along the row, and then moving down into the next row. In the left hand diagram we try to remove a horizontal strip first, and then a vertical strip. In the right hand diagram we try to remove a vertical strip first, and then a horizontal strip. Either of these approaches provides a maximal solution.

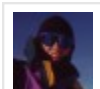
If instead of requiring the 3×1 strip to be straight, we wanted to cut L-shaped pieces, then we can cut 19 pieces with no wastage.



[Reply](#)

Jim Randell 16 November 2019 at 12:42 pm

The published solution includes a neat way to show there cannot be more than 15 strips.



If we label the squares “a”, “b”, “c”, starting at the top right, and working across the rows, we get a layout like this:

a	b	c	a	b	c	a	b
c	a	b	c	a	b	c	a
b	c	a	b	c	a	b	c
a	b	c	a	b	c	a	b
c	a	b	c	a	b	c	a
b	c	a	b	c	a	b	c
a	b	c	a	b	c	a	b
c	a	b	c	a	b	c	a

Note how the categories run along the diagonals.

Each horizontal or vertical strip of 3 squares must use exactly 1 square from each category.

The removed squares are all category “a”, which leaves only 15 “a” squares on the board (and there are 21 “b” and 21 “c” squares), so we cannot make more than 15 strips.

Of course we still need to show that we can make 15 strips, but as detailed above a greedy algorithm immediately gives us a way to cut 15 strips, so there is no need to investigate further.

[Reply](#)

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)

↑ Top