pb45

March 8, 2020

[183]:

```
from itertools import islice, count
import math
import matplotlib.pyplot as plt
```

[184]:

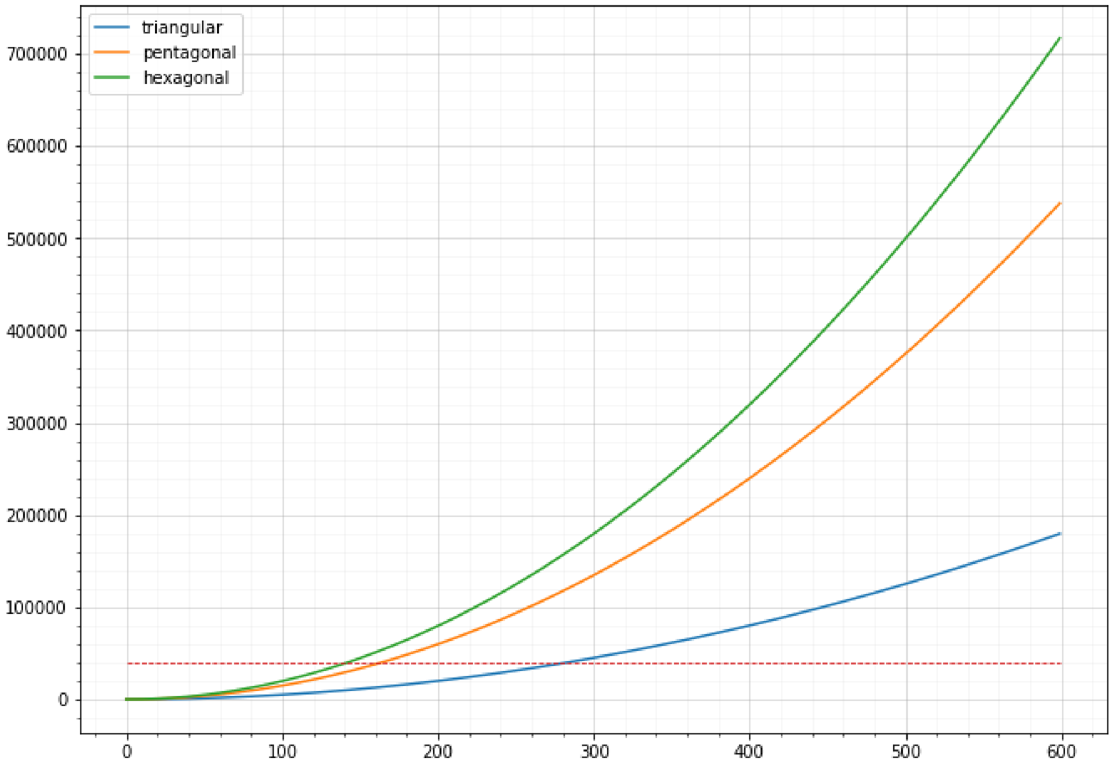
```
length = lambda n : math.ceil(math.log(n, 10)) # find the length of an int
```

[185]:

```
plt.figure(figsize=(11,8))
n = 600
x = list((40755 for _ in range(n)))
y1,y2,y3 = zip(*((n*(n+1)//2, n*(3*n-1)//2, n*(2*n-1)) for n in range(n)))
plt.plot(y1, label='triangular') ; plt.plot(y2, label='pentagonal') ; plt.
 ↪plot(y3, label='hexagonal')
plt.minorticks_on()
plt.grid(which='major', alpha=0.5) ; plt.grid(which='minor', alpha=0.1)
plt.plot(x, linewidth=0.8, linestyle='--') ; plt.legend()
```

[185]:

```
<matplotlib.legend.Legend at 0x7f3e775f6e10>
```



[186]:
```
is_triangular = lambda n : (0.5*(-1+(1+8*n)**0.5)).is_integer()
is_pentagonal = lambda n : ((1+(1+24*n)**0.5)/6).is_integer()
is_hexagonal = lambda n : (0.25*(1+(1+8*n)**0.5)).is_integer()
```

[187]:
```
def numbers():
    n = 1
    while True:
        hexa = n*(2*n-1)
        if is_triangular(hexa) and is_pentagonal(hexa):
            yield hexa
        n += 1
```

[188]:
```
list(islice(numbers(), 3))
```

[188]:
```
[1, 40755, 1533776805]
```

[189]:
```
list(islice((hexa for hexa in (n*(2*n-1) for n in count(1)) if␣
 ↪is_triangular(hexa) and is_pentagonal(hexa)), 3))
```

[189]:
```
[1, 40755, 1533776805]
```

[190]:
```
%timeit list(islice(numbers(), 3))
```

```
18.4 ms ± 856 µs per loop (mean ± std. dev. of 7 runs, 100 loops each)
```

[191]:

```
%timeit list(islice((hexa for hexa in (n*(2*n-1) for n in count(1)) if is_triangular(hexa) and is_pentagonal(hexa)), 3))
```

```
19.8 ms ± 613 µs per loop (mean ± std. dev. of 7 runs, 100 loops each)
```