project_euler_1_to_10

April 8, 2020

1 Pb1

[1]:
```
sum((i for i in range(1000) if i%3 == 0 or i%5 == 0))
```

[1]:
```
233168
```

2 Pb2

[2]:
```
s = 0
u, v = 1, 2
while v <= 4_000_000:
    if v%2 == 0:
        s += v
    temp = u
    u = v
    v += temp
print(s)
```

```
4613732
```

3 Pb3

[3]:
```
def is_prime(n):
    if n==2: return True
    if n%2 == 0: return False
    for i in range(3, int(n**0.5)+1, 2):
        if n%i == 0: return False
    return True
```

[4]:
```
print([i for i in range(1,21) if is_prime(i)])
```

```
[1, 2, 3, 5, 7, 11, 13, 17, 19]
```

[5]:

```
def divisors(n):
    mx = 0
    for i in range(int(n**0.5), 1, -1):
        if n%i == 0:
            temp = n//i if is_prime(n//i) else i if is_prime(i) else 0
            if temp >= mx:
                mx = temp
    return mx
```

[6]:

```
divisors(600851475143)
```

[6]:

```
6857
```

4 Pb4

[7]:

```
is_palindrome = lambda astr : astr == astr[::-1]
```

[8]:

```
tests = ['121', '12']
print('\n'.join(('is palindrome %s -> %s' % (test, is_palindrome(test)) for␣
 ↪test in tests)))
```

```
is palindrome 121 -> True
is palindrome 12 -> False
```

[9]:

```
max((i*j for i in range(100,1000) for j in range(100, 1000) if␣
 ↪is_palindrome(str(i*j))))
```

[9]:

```
906609
```

5 Pb5

[10]:

```
def multiplication(liste):
    if len(liste) == 1:
        return liste[0]
    return liste[0] * multiplication(liste[1:])
```

[11]:

```
multiplication([2,3,2,5,7,2,3])
```

[11]:

```
2520
```

[12]:

```
def pb5(lim):
    liste = [i for i in range(2,lim+1)]
    for i in range(lim-2):
        for j in range(i+1, lim-1):
            if liste[j]%liste[i] == 0:
                liste[j] /= liste[i]
    return int(multiplication(liste))
pb5(20)
```

```
[12]: 232792560
```

6 Pb6

```
[13]: def diff(lim):
          s = lim*(lim+1)*0.5
          s *= s
          return int(s - sum((i**2 for i in range(1, lim+1))))
```

```
[14]: diff(100)
```

```
[14]: 25164150
```

7 Pb7

```
[15]: from itertools import islice, takewhile
```

```
[16]: def primes():
          p = 2
          while True:
              if is_prime(p):
                  yield p
              p += 1
```

```
[17]: list(islice(primes(), 10001))[-1]
```

```
[17]: 104743
```

8 Pb8

```
[18]: with open('1000_digits.txt', 'r') as f:
          number = [line for line in f.readlines()]
      f.close()
```

```
[19]: number = (''.join(number)).replace('\n', '')
```

[20]:

```
def mult(astr):
    res = 1
    for item in astr:
        res *= int(item)
    return res
```

[21]:

```
def product(number, adj=4):
    return max((mult(number[i:i+adj]) for i in range(len(number)-adj+1)))
product(number)
```

[21]:

```
5832
```

[22]:

```
product(number, 13)
```

[22]:

```
23514624000
```

9 Pb9

[23]:

```
def res():
    for c in range(1001):
        for b in range(1001-c):
            for a in range(1001-b-c):
                if a + b + c == 1000:
                    if a*a + b*b == c*c:
                        return a*b*c
res()
```

[23]:

```
31875000
```

[24]:

```
%timeit res
```

```
34.4 ns ± 0.0937 ns per loop (mean ± std. dev. of 7 runs, 10000000 loops each)
```

10 Pb10

[25]:

```
def primes(n):
    prime = [True for _ in range(n+1)]
    p = 2
    while p*p <= n:
        if prime[p]:
            for i in range(p*2, n+1, p):
                prime[i] = False
        p += 1
    prime[0] = False
    prime[1] = False
    return prime
```

[26]:

```
print([i for i, p in enumerate(primes(21)) if p])
```

```
[2, 3, 5, 7, 11, 13, 17, 19]
```

[27]:

```
sum((i for i, p in enumerate(primes(2_000_000)) if p))
```

[27]:

```
142913828922
```