```
import time
def timeit(f):
    def wrapper(*args):
        t = time.time()
        r = f(*args)
        t = time.time() - t
        print (t)
        return r
    return wrapper

def factorize_naive(n):
    factors = []
    p = 2
    while True:
        if n == 1:
            return factors
        if n % p == 0:
            factors.append(p)
            n = n / p
        elif p * p >= n:                 # n is prime now
            factors.append(n)
            return factors
        elif p > 2: # Advance in steps of 2 over odd numbers
            p += 2
        else:       # If p == 2, get to 3
            p += 1
    assert False, "unreachable"

def factorize_smart(n):
    factors = []
    f = 2
    while n % f == 0:
        factors.append(f)
        n /= f
    f = 3
    while f * f <= n:
        while n % f == 0:
            factors.append(f)
            n /= f
        f += 2
    if n > 1:
        factors.append(n)
    return factors

def factorize_recursive(n):
    if n < 2:
        return list()
    if n % 2 == 0:
        return [2] + factorize_recursive(n/2)
    for f in range(3, n+1, 2):
        if n % f == 0:
            return [f] + factorize_recursive(n/f)
    assert 0, n

def product(ii):
    p = 1
    for i in ii:
        p *= i
    return p
```

```python
@timeit
def test_factorize(f, max, check=False):
    count = 0
    for n in range(2, max):
        ff = f(n)
        count += len(ff)
        if check:
            assert product(ff) == n
    return count

if __name__ == '__main__':

    # for i in range(2,1000):
    #     ff = factorize_recursive(i)
    #     assert product(ff) == i
    #     ff = factorize_naive(i)
    #     assert product(ff) == i
    #     # print (i, ff)

    print("")
    max = 1000
    for check in [0,1]:
        a = test_factorize(factorize_smart, max, check)
        b = test_factorize(factorize_naive, max, check)
        c = test_factorize(factorize_recursive, max, check)
        assert a == b == c

    for f in factorize_smart, factorize_naive, factorize_recursive:
        for i in range(2, 10):
            print(f(i)),
        print("")
```