**Kernel:** SageMath 10.7

# Counting Primes

MATH 242 Modern Computational Mathematics

First, here is a copy of our sieve of Eratosthenes from the previous class.

In [6]:
```
def sieveEratosthenes(nMax):
    # initialize a list
    # if you start the list from zero, then each number in the list is
EQUAL to its index in the list, which is very convenient
    nums = list(range(0, nMax + 1))

    # we will replace non-prime numbers in the list with zeros
    # start by replacing 1 with 0
    nums[1] = 0

    # loop over list items until we reach sqrt(nMax)
    i = 2
    while i <= sqrt(nMax):
        # if nums[i] is not zero, then it is prime
        if nums[i] > 0:
            # replace all multiples of nums[i] with zeros
            for j in range(2*i, nMax + 1, i):
                nums[j] = 0

        # testing: print the current state of nums
        # print(f"finished i={i}: nums={nums}\n")

        # increment i
        i += 1

    # return a list containing all nonzero elements of nums
    return [i for i in nums if i != 0]
```

## The Prime Counting Function

Define the **prime counting function** $\pi(x)$ to be the number of primes less than or equal to any real number $x$. For example, $\pi(10) = 4$ since there are 4 primes less than or equal to 10: specifically, 2, 3, 5, and 7.

Perhaps the simplest way to compute $\pi(x)$ is to find the length of the list returned by `sieveEratos(x)` .

In [10]:
```
def pi(x):
    return len(sieveEratosthenes(floor(x)))
```

For example:

In [7]:
```
pi(100)
```

Out[7]: 25

In [11]:
```
pi(100.4)
```

Out[11]: 25

We can then plot values of the prime counting function. First we need to make a list of values of $\pi(x)$.

In [15]:
```
nMax = 1000
piVals = [pi(n) for n in range(2, nMax)]
print(piVals)
```

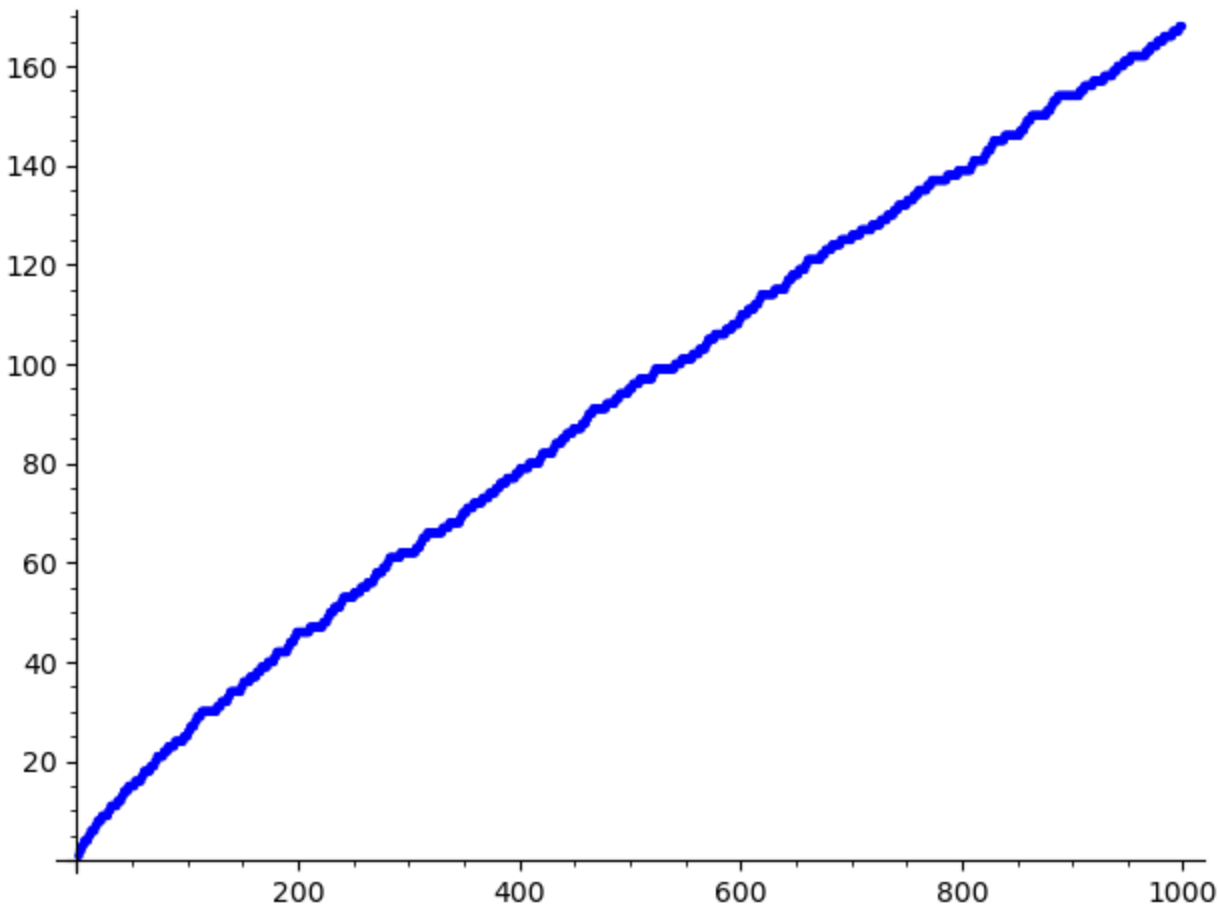
Out[15]:
```
[1, 2, 2, 3, 3, 4, 4, 4, 4, 5, 5, 6, 6, 6, 6, 7, 7, 8, 8, 8, 8, 9, 9, 9, 9,
9, 9, 10, 10, 11, 11, 11, 11, 11, 11, 12, 12, 12, 12, 13, 13, 14, 14, 14,
14, 15, 15, 15, 15, 15, 15, 16, 16, 16, 16, 16, 16, 17, 17, 18, 18, 18, 18,
18, 18, 19, 19, 19, 19, 20, 20, 21, 21, 21, 21, 21, 21, 22, 22, 22, 22, 23,
23, 23, 23, 23, 23, 24, 24, 24, 24, 24, 24, 24, 24, 25, 25, 25, 25, 26, 26,
27, 27, 27, 27, 28, 28, 29, 29, 29, 29, 30, 30, 30, 30, 30, 30, 30, 30, 30,
30, 30, 30, 30, 30, 31, 31, 31, 31, 32, 32, 32, 32, 32, 32, 33, 33, 34, 34,
34, 34, 34, 34, 34, 34, 34, 34, 35, 35, 36, 36, 36, 36, 36, 36, 37, 37, 37,
37, 37, 37, 38, 38, 38, 38, 39, 39, 39, 39, 39, 39, 40, 40, 40, 40, 40, 40,
41, 41, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 43, 43, 44, 44, 44, 44, 45,
45, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 47, 47, 47, 47, 47, 47,
47, 47, 47, 47, 47, 47, 48, 48, 48, 48, 49, 49, 50, 50, 50, 50, 51, 51, 51,
51, 51, 51, 52, 52, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 54, 54, 54, 54,
54, 54, 55, 55, 55, 55, 55, 55, 56, 56, 56, 56, 56, 56, 57, 57, 58, 58, 58,
58, 58, 58, 59, 59, 59, 59, 60, 60, 61, 61, 61, 61, 61, 61, 61, 61, 61, 61,
62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 63, 63, 63, 63, 64,
64, 65, 65, 65, 65, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66,
67, 67, 67, 67, 67, 67, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 69, 69, 70,
70, 70, 70, 71, 71, 71, 71, 71, 71, 72, 72, 72, 72, 72, 72, 72, 72, 73, 73,
73, 73, 73, 73, 74, 74, 74, 74, 74, 74, 75, 75, 75, 75, 76, 76, 76, 76, 76,
76, 77, 77, 77, 77, 77, 77, 77, 77, 78, 78, 78, 78, 79, 79, 79, 79, 79, 79,
79, 79, 80, 80, 80, 80, 80, 80, 80, 80, 80, 80, 81, 81, 82, 82, 82, 82, 82,
82, 82, 82, 82, 82, 83, 83, 84, 84, 84, 84, 84, 84, 85, 85, 85, 85, 86, 86,
86, 86, 86, 86, 87, 87, 87, 87, 87, 87, 87, 87, 88, 88, 88, 88, 89, 89, 90,
90, 90, 90, 91, 91, 91, 91, 91, 91, 91, 91, 91, 91, 91, 91, 92, 92, 92, 92,
92, 92, 92, 92, 93, 93, 93, 93, 94, 94, 94, 94, 94, 94, 94, 94, 95, 95, 95,
95, 96, 96, 96, 96, 96, 96, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97,
98, 98, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99,
99, 100, 100, 100, 100, 100, 100, 101, 101, 101, 101, 101, 101, 101, 101,
101, 101, 102, 102, 102, 102, 102, 102, 103, 103, 103, 103, 103, 103, 104,
104, 105, 105, 105, 105, 105, 105, 106, 106, 106, 106, 106, 106, 106, 106,
106, 106, 107, 107, 107, 107, 107, 107, 108, 108, 108, 108, 108, 108, 109,
109, 110, 110, 110, 110, 110, 110, 111, 111, 111, 111, 111, 111, 112, 112,
112, 112, 113, 113, 114, 114, 114, 114, 114, 114, 114, 114, 114, 114, 114,
114, 115, 115, 115, 115, 115, 115, 115, 115, 115, 115, 116, 116, 117, 117,
117, 117, 118, 118, 118, 118, 118, 118, 119, 119, 119, 119, 119, 119, 120,
120, 121, 121, 121, 121, 121, 121, 121, 121, 121, 121, 121, 121, 122, 122,
122, 122, 123, 123, 123, 123, 123, 123, 124, 124, 124, 124, 124, 124, 124,
124, 125, 125, 125, 125, 125, 125, 125, 125, 125, 125, 126, 126, 126, 126,
126, 126, 126, 126, 127, 127, 127, 127, 127, 127, 127, 127, 127, 127, 128,
128, 128, 128, 128, 128, 128, 128, 129, 129, 129, 129, 129, 129, 130, 130,
130, 130, 130, 130, 131, 131, 131, 131, 132, 132, 132, 132, 132, 132, 132,
132, 133, 133, 133, 133, 133, 133, 134, 134, 134, 134, 135, 135, 135, 135,
135, 135, 135, 135, 136, 136, 136, 136, 137, 137, 137, 137, 137, 137, 137,
```

```
137, 137, 137, 137, 137, 137, 137, 138, 138, 138, 138, 138, 138, 138, 138,
138, 138, 139, 139, 139, 139, 139, 139, 139, 139, 139, 139, 139, 140,
140, 141, 141, 141, 141, 141, 141, 141, 141, 141, 142, 142, 143, 143,
143, 143, 144, 144, 145, 145, 145, 145, 145, 145, 145, 145, 145, 145, 146,
146, 146, 146, 146, 146, 146, 146, 146, 146, 146, 146, 146, 147, 147,
147, 147, 148, 148, 149, 149, 149, 149, 150, 150, 150, 150, 150, 150, 150,
150, 150, 150, 150, 150, 150, 150, 151, 151, 151, 151, 152, 152, 153, 153,
153, 153, 154, 154, 154, 154, 154, 154, 154, 154, 154, 154, 154, 154, 154,
154, 154, 154, 154, 154, 154, 154, 155, 155, 155, 155, 156, 156, 156, 156,
156, 156, 156, 156, 157, 157, 157, 157, 157, 157, 157, 157, 157, 157, 158,
158, 158, 158, 158, 158, 158, 158, 159, 159, 159, 159, 160, 160, 160, 160,
160, 160, 161, 161, 161, 161, 161, 161, 162, 162, 162, 162, 162, 162, 162,
162, 162, 162, 162, 162, 162, 162, 163, 163, 163, 163, 164, 164, 164, 164,
164, 164, 165, 165, 165, 165, 165, 165, 166, 166, 166, 166, 166, 166, 166,
166, 167, 167, 167, 167, 167, 167, 168, 168, 168]
```

Now we can plot the prime counting function.

In [16]:
```
list_plot( list(zip(range(2,nMax), piVals)) )
```

Out[16]:




Unfortunately, this is inefficient because we are running the sieve of Eratosthenes for each individual data point above.

We should be able to compute a single list of primes up to $N$ and get all of the counts from that list. Let's do that in the next code cell:

In [21]:
```
# returns a list of values of the prime-counting function π(x), for
integers x from 1 to nMax
def computePiVals(nMax):
    # compute a list of primes up to nMax
    primes = sieveEratosthenes(nMax)

    # make a list of nMax+1 zeros
    piVals = [0]*(nMax + 1)

    # track how many primes we've found so far
    count = 0

    # loop over integers i from 2 to nMax
    for i in range(2, nMax + 1):
        # if i is the next prime, then add 1 to our count
        #print(count)
        if count < len(primes) and i == primes[count]:
            count += 1

        # store the current count in piVals[i]
        piVals[i] = count

    # return the list of piVals
    return piVals
```

Try out this new function:

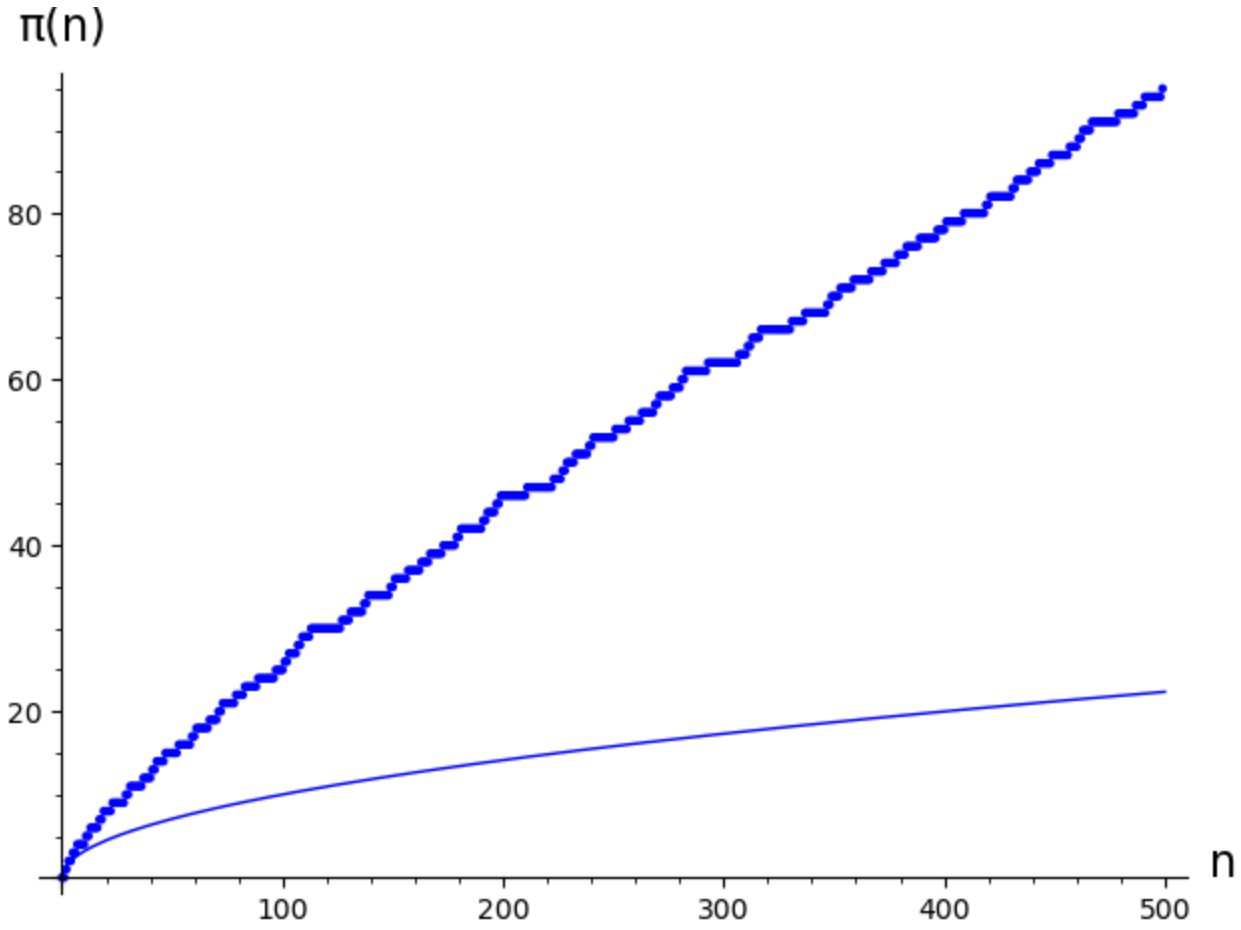
In [23]:
```
print( computePiVals(100) )
```

Out[23]:
```
[0, 0, 1, 2, 2, 3, 3, 4, 4, 4, 4, 5, 5, 6, 6, 6, 6, 7, 7, 8, 8, 8, 8, 9, 9,
9, 9, 9, 9, 10, 10, 11, 11, 11, 11, 11, 11, 12, 12, 12, 12, 13, 13, 14, 14,
14, 14, 15, 15, 15, 15, 15, 15, 16, 16, 16, 16, 16, 16, 17, 17, 18, 18, 18,
18, 18, 18, 19, 19, 19, 19, 20, 20, 21, 21, 21, 21, 21, 21, 22, 22, 22, 22,
23, 23, 23, 23, 23, 23, 24, 24, 24, 24, 24, 24, 24, 24, 25, 25, 25, 25]
```

Now we can quickly compute a huge list of values of $\pi(x)$ and make a plot.

In [30]:
```
nMax = 500
piVals = computePiVals(nMax)
plot1 = list_plot( list(zip(range(nMax), piVals)), axes_labels=
["n","π(n)"])
plot2 = plot(sqrt(x), [x,0,nMax])
plot1 + plot2
```

Out[30]:




## Exploration

Discuss with your group the following questions:

1. What is the shape of the graph of $\pi(x)$? Can you find a simple function that approximates $\pi(x)$?
2. What proportion of the first $N$ positive integers are prime? How does this depend on $N$?
3. Use your best answers to the previous questions to the previous questions, how many primes do you think are less than $10^{20}$? How many primes do you think are less than $10^{100}$?

In [0]:

In [0]: