

Day 31

```
In [2]: import numpy as np
import matplotlib.pyplot as plt
```

```
In [3]: def bisection(function, lower_guess, upper_guess, tolerance=2**-32):
midpoint = (lower_guess + upper_guess)/2
while upper_guess - lower_guess > tolerance:
    if function(lower_guess)*function(midpoint)<0:
        upper_guess = midpoint
        midpoint = (lower_guess + upper_guess)/2
    elif function(midpoint)*function(upper_guess)<0:
        lower_guess = midpoint
        midpoint = (lower_guess + upper_guess)/2
    elif function(lower_guess)*function(midpoint)>0 and function(midpoint)*function(upper_guess)>0:
        print('no unique root in that bracket')
        break
    return(midpoint)
```

```
In [8]: fig0, ax0 = plt.subplots()
x = np.linspace(-1,5,100)
ax0.plot(x, 5*np.exp(-x)+x-5)
```

Out[8]: [<matplotlib.lines.Line2D at 0x7f3a48f0b400>]

```
In [9]: def f(x):
return(5*np.exp(-x)+x-5)
```

```
In [10]: bisection(f, 4, 6)
```

Out[10]: 4.9651142318034545

```
In [15]: def newton(f, df, guess, tolerance = 2**-32):
x = guess
n = 0
while abs(f(x)) > tolerance:
    x = x - f(x)/df(x)
    n += 1
return(x, n)
```

```
In [12]: def df(x):
return(-5*np.exp(-x)+1)
```

```
In [29]: newton(f, df, 9)
```

Out[29]: (4.965114231750048, 3)

```
In [ ]: def secant(f, guess, delta, tolerance = 2**-32):
x0 = guess
x1 = x0 + delta
n = 0
while abs(f(x1))>tolerance:
```