In [9]:

```
#newton_raphson method
import math
import timeit


def f(x):
    return x**6 - x*2 -5
def df(x):
    return 6*x**5 - 2

x0=1
tolerance=1e-6
max_iteration = 100
count =0

def newton_raphson_method():
    global x0, count,tolerance,max_iteration
    for i in range (max_iteration):
        fx=f(x0)
        dfx =df(x0)
        x1 =x0 -f(x0)/df(x0)
        count+=1
        if abs (f(x1)) < tolerance:
            print(f"Root found at x ={x1:.6f}")
            break
        else:
            x0 = x1

# Measure the execution time
execution_time = timeit.timeit(newton_raphson, number=1)

# Print the execution time
print(f"Execution time: {execution_time:.5f} secs")


print(f"count =: {count}")
```

```
Root found at x =1.408787
Execution time: 0.00020 secs
count =: 8
```

In [10]:

```
#bisection method
import math

def f(x):
    return x**4-3**x-8

a=2
b=3
tolerance = 1e-6
max_iteration = 100

for i in range (max_iteration):
    c=(a+b)/2
    if abs(f(c)) < tolerance:
        print(f"Root found at x={c:.6f}")
        break
    elif f(c)*f(a)<0:
        b=c
    else:
        a=c
```

```
Root found at x=2.043604
```

In [ ]: