APPLICATION PERFORMANCE METRICS

Assignment Date	18 november 2022
Student Name	G.karthika
Student Roll Number	815819104011
Maximum Marks	2 Marks

Application performance

```
tracing_start()

start = time.time()

sq_list1 = [elem + elem**2 for elem in range(1,1000)]

#print(sq_list1)

end = time.time()

print("time elapsed {} milli seconds".format((end-start)*1000))

tracing_mem()

#Result

Tracing Status :  False

Tracing Status :  True

time elapsed 7.999420166015625 milli seconds

Peak Size in MB -  0.04634556579589844
```

Method – 1

```
tracing_start()

start = time.time()

list_word =

["Quantify","performance","improvements","in","Python"]

s = ""

for substring in list_word:

    s += substring + " "

print(s)

end = time.time()

print("time elapsed {} milli seconds".format((end-start)*1000))

tracing_mem()

#Result

Tracing Status :  False

Tracing Status :  True
```

```
time elapsed 0.0 milli seconds

Peak Size in MB -  0.015248298645019531
```

Method – 1

```
tracing_start()

start = time.time()

a = [2,3,3,2,5,4,4,6,5,7,7,3,3,4,7,2,5,2,5]

b = []

for i in a:

    if i not in b:

        b.append(i)

print(b)

end = time.time()

print("time elapsed {} milli seconds".format((end-start)*1000))

tracing_mem()

#Result

Tracing Status :  False

Tracing Status :  True
```

```
[2, 3, 5, 4, 6, 7]

time elapsed 0.0 milli seconds

Peak Size in MB -  0.0144805908203125
```

method-2

```
tracing_start()

tracing_start()

start = time.time()

a = [2,3,3,2,5,4,4,6,5,7,7,3,3,4,7,2,5,2,5]

set_list = list(set(a))

print(set_list)

end = time.time()

print("time elapsed {} milli seconds".format((end-start)*1000))

tracing_mem()

#Result

Tracing Status :  False

Tracing Status :  True

[2, 3, 5, 4, 6, 7]
```

```
time elapsed 0.0 milli seconds

Peak Size in MB -  0.012772964477539062
```

The first one is using pandas and the second is using dask. Dask is a flexible library for parallel computing in Python. The documentation can be found at .

```
tracing_start()

start = time.time()

df = pd.read_csv("F:/Av_Hackathon/DataScienceJobs/ds_jobs.csv")

df_by_discipline = df.groupby('major_discipline').count()[['enrollee_id']]

end = time.time()

print("time elapsed {} milli seconds".format((end-start)*1000))

tracing_mem()

#Result

Tracing Status :  False

Tracing Status :  True

time elapsed 136.00635528564453 milli seconds
```