

## CMPSC 412 – Lab-1 (25 points)

Due date: 9/6/2022

### Lab Exercises:

Design a program to generate a specified quantity of integers (minimum 1000 integers). Use rand function to generate integers between 1 and 10000. Note that these random numbers can contain duplicates. Store this collection of integers in a) Lists and b) Dictionaries. Perform the following for the List and the Dictionary:

1. Print all the elements. Perform this operation 3 times for each data structure. Measure empirically the time it takes to print all the elements each time. Tabulate your results – Table-1.

#sample of results – screenshots and Table-1 as shown below

```
list_time = [0.0018759360000331071, 0.0021980430001349305, 0.002879930999370117]  
dict_time = [0.0020998019999751705, 0.0022940429998925538, 0.0018259250000483007]
```

Print operation	List	Dictionary
Trail-1	0.0018759360000331071	0.0020998019999751705
Trail-2	0.0021980430001349305	0.0022940429998925538
Trail-3	0.002879930999370117	0.0018259250000483007

2. A series of retrievals of random values (use rand function) in the collection and measure empirically the time it takes to do the retrievals. Note: use the same value generated from rand function in list and dictionary. You should run this program 3 times for each data structure. Tabulate your results – Table-2.

```
list_time = [0.0067651490001026104, 0.004936253000096258, 0.005681665999873076]  
dict_time = [0.0028195879999657336, 0.0023815310000827594, 0.002407330000096408]
```

#sample of results – screenshots and Table-2 as shown below

Find operation	List	Dictionary
Trail-1	0.0067651490001026104	0.0028195879999657336
Trail-2	0.004936253000096258	0.0023815310000827594
Trail-3	0.005681665999873076	0.002407330000096408

3. Generate a random number, perform insertion operation and print all the elements. You should run this program 3 times for each data structure. Tabulate your results – Table-3.

```
list_time = [0.0043847909998930845, 0.004129273999978977, 0.004836165999904551]
dict_time = [0.0021348829998260044, 0.002605042999903162, 0.00306323900008465]
```

#sample of results – screenshots and Table-3 as shown below

Insert operation	List	Dictionary
Trail-1	0.0043847909998930845	0.0021348829998260044
Trail-2	0.004129273999978977	0.002605042999903162
Trail-3	0.004836165999904551	0.00306323900008465

4. Generate a random number and perform deletion operation on the list and dictionary data structures. You should run this program 3 times for each data structure. Tabulate your results – Table-4.

```
list_time = [0.00011957900005654665, 0.00014614999963669106, 0.00015715300014562672]
dict_time = [0.00011068300000260933, 0.000176021000243054, 0.00017535200004203944]
```

#sample of results – screenshots and Table-4 as shown below

Delete operation	List	Dictionary
Trail-1	0.0001195790000565466	0.00011068300000260933
Trail-2	0.0001461499996366910	0.000176021000243054
Trail-3	0.0001571530001456267	0.00017535200004203944

## **Conclusion:**

What do you understand from this exercise? Explain in detail.

I understand that for searching a specific data using a dictionary is faster, it is most likely because dictionary implements hashmap, it don't depending on iterating over all the values to find what you want, which makes searching for data have a time complexity of  $O(1)$ .

List on the other hand need to be iterated to look for a specific value which is why it have a time complexity of  $O(n)$  which make it slower than dictionary for looking up.

### Deliverables:

Report and the program file.

A sample method to get time measurement:

```
long startTime;  
long stopTime;  
...  
startTime = System.currentTimeMillis();  
//do whatever you want to measure  
stopTime = System.currentTimeMillis();  
System.out.println("Elapsed time = "+(stopTime-startTime)+" msecs.");
```