

Lab_1_v2

Custom Functions

```
def getDuration(start: time.time(), end: time.time()):
    return end - start

def getUniqueRandomNumbers(count: int, min:int, max:int):
    vals = []
    r = random.randint(min, max)
    for i in range(count):
        while r in vals:
            r = random.randint(min, max)
        vals.append(r)
    return vals

def getUniqueRandomDictKeys(count: int, d: dict):
    vals = []
    r = random.randint(0, len(d.keys()))
    for c in range(count):
        while r in vals or r not in d.keys():
            r = random.randint(0, len(d.keys()))
        vals.append(list(d.keys())[r])
    return vals

def printResults(title: str, list_times: [], dict_times: []):
    avg_list = sum(list_times)/len(list_times)
    avg_dict = sum(dict_times)/len(dict_times)
    print("\n"+title)
    print("\tList Times: "+str(list_times) + " | Average:
"+str(avg_list))
    print("\tDict Times: " + str(dict_times) + " Average:
"+str(avg_dict))

    if avg_list < avg_dict:
        print("\t\tFastest: List")
    else:
        print("\t\tFastest: Dictionary")
```

Generating List and Dictionary

```
test_dict = dict()
test_list=[]

for i in range(1000000):
    x=random.randint(1,100000)
    test_dict[i] = x
    test_list.append(x)
```

Functions

Exercise 1

```
"Print Testing"
list_print_times = []
dict_print_times = []

for i in range(3):
    start = time.time()
    print(test_list)
    end = time.time()
    list_print_times.append(getDuration(start,end))

    start = time.time()
    print(test_dict)
    end = time.time()
    dict_print_times.append(getDuration(start, end))
```

Exercise 2

```
""""Retrieval Testing""""
list_retrieval_times = []
dict_retrieval_times = []

for i in range(3):
    r = random.randint(0,len(test_list))
    start=time.time()
```

```

    exists = r in test_list #performs a retrieval and stores existence of
the object in the list as a boolean
    end=time.time()

    list_retrieval_times.append(getDuration(start, end))
    start = time.time()
    exists = r in test_dict # performs a retrieval and stores existence
of the object in the list as a boolean
    end = time.time()
    dict_retrieval_times.append(getDuration(start, end))

```

Exercise 3

```

"""Insertion"""
list_insertion_times = []
dict_inssertion_times = []

for i in range(3):
    r = random.randint(0, len(test_list))
    start=time.time()
    test_list.insert(r,r)
    end=time.time()
    list_insertion_times.append(getDuration(start,end))

    start = time.time()
    test_dict[r] = r
    end = time.time()
    dict_inssertion_times.append(getDuration(start, end))

```

Exercise 4

```

#generating 3 unique random numbers to avoid removing items more than
once
list_remove_indecies = getUniqueRandomNumbers(3, 1, len(test_list))
dict_remove_keys = list_remove_indecies
print(list_remove_indecies)

```

```

list_remove_times = []
dict_remove_times = []

"Deletion"

for j in list_remove_indecies:
    start=time.time()
    del test_list[j]
    end=time.time()
    list_remove_times.append(getDuration(start,end))

    start = time.time()
    test_dict.__delitem__(j)
    end = time.time()
    dict_remove_times.append(getDuration(start, end))

```

```

"output section"
print("\n----- Results -----
-----")
printResults("Printing", list_print_times, dict_print_times)
printResults("Retrieval", list_print_times, dict_print_times)
printResults("Insertion", list_insertion_times, dict_inssertion_times)
printResults("Deletion", list_remove_times, dict_remove_times)

```

Output

```

----- Results -----
----
Printing
      List Times: [0.1437211036682129, 0.20246124267578125,
0.2051541805267334] | Average: 0.18377884229024252
      Dict  Times: [0.3435940742492676, 0.4738330841064453,
0.27918529510498047] | Average: 0.36553748448689777
                Fastest: List

Retrieval
      List Times: [0.1437211036682129, 0.20246124267578125,
0.2051541805267334] | Average: 0.18377884229024252

```

Dict Times: [0.3435940742492676, 0.4738330841064453,
0.27918529510498047] | Average: 0.36553748448689777

Fastest: List

Insertion

List Times: [0.00028014183044433594, 0.0002129077911376953,
0.002937793731689453] | Average: 0.0011436144510904949

Dict Times: [9.5367431640625e-07, 1.1920928955078125e-06,
1.1920928955078125e-06] | Average: 1.1126200358072917e-06

Fastest: Dictionary

Deletion

List Times: [0.0003409385681152344, 0.0003199577331542969,
6.67572021484375e-05] | Average: 0.0002425511678059896

Dict Times: [3.0994415283203125e-06, 1.9073486328125e-06,
9.5367431640625e-07] | Average: 1.986821492513021e-06

Fastest: Dictionary