Introduction

I wrote a program that sorts passwords in a linked list then

prints the top 20 most used passwords

Proposed solution

At first, I exported my file into a utf-8 file to avoid in wacky characters in code. I then made my node class to be able to create a linked list and placed the passwords in the linked list. I then wrote a Boolean function to check if a password was already in the linked list and if it was it would to the count and would not be put in the linked list. I then followed the merge sort algorithm (finding the pivot, then splitting the linked list in half then sorting the split linked lists then adding them all together). I then used python’s collections library to be able to print out the top 20 used passwords. In order to successfully do this though I had to put the linked list in a python list.

Experimental results

The test cases took a very long time as the we had 10 million passwords to work with, to deal with this I condensed the tests cases to 10 passwords in a file and tested my code there.

Sample File.txt :

atorges rag2k3xi

atorgon 251200at

ATORGUY HOUNDDOG

a\_tori 3225177

atori36 policy

atorichko sasha

atorii at7856

atorikian anechka

atorin92 w5111711

atorina atorina

SAMPLE #1 Merge Sort Implantation



SAMPLE #2 Bubble Sort Implantation

Sample File.txt :

atorges rag2k3xi

atorgon 251200at

ATORGUY HOUNDDOG

a\_tori 3225177

atori36 policy

atorichko sasha

atorii at7856

atorikian anechka

atorin92 w5111711

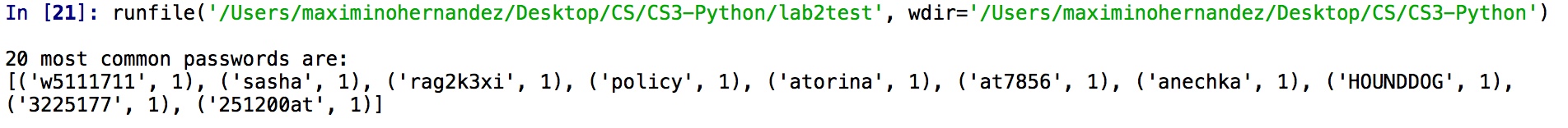
atorina atorina

atorina atorina

atorina atorina



SAMPLE #3 Prints most popular passwords (same sample file as above)



•Conclusions

I learned how to recursively create a linked list in python and how to implement merge sort on a linked list on python.

The time complexity of this program would be O(n^2)

Linked list search takes O(n^2) with bubble sort time while merge sort is O(nlogn) time. The top 20 list takes O(n) time.

•Appendix

|  |  |
| --- | --- |
|  | # -\*- coding: utf-8 -\*- |
|  | #Jesus Maximino Hernandez 88756947 |
|  | #Data Structures - Diego Aguirre |
|  | #Lab 2 - Option B |
|  | #Program sorts passwords in linked list then |
|  | #prints the top 20 most used passwords |
|  | """ |
|  | Created on Wed Oct 15 20:57:02 2018 |
|  |  |
|  | @author: JesusMHernandez |
|  | """ |
|  |  |
|  | import collections |
|  | #creates node |
|  | class Node: |
|  | password ="" |
|  | count = 1 |
|  | next = None |
|  | def \_\_init\_\_(self, password, count, next): |
|  | self.password = password |
|  | self.next = next |
|  | def \_\_in\_\_(self, password, next): |
|  | self.password = password |
|  | self.next = next |
|  |  |
|  | Node(password, 1, temp) |
|  |  |
|  | #function to print linked list |
|  | def printList(llist): |
|  | while llist is not None: |
|  | #Prints the number of times the linked list comes out |
|  | #print(llist.count) |
|  | print(llist.password) |
|  | llist = llist.next |
|  | #check if password is alread in list |
|  | def inList(llist, password): |
|  | tempPass = llist |
|  | while tempPass is not None: |
|  | if(tempPass.password == password): |
|  | tempPass.count += 1 |
|  | #print(tempPass.count) |
|  |  |
|  | tempPass = tempPass.next |
|  | return False |
|  | #Prints top 20 most popular passwords |
|  | def popular(llist): |
|  | temp = llist |
|  | tempList = [] |
|  | #gets elements from linked list and puts it into a list |
|  | while llist is not None: |
|  | tempList.append(temp.password) |
|  | temp = temp.next |
|  | llist = llist.next |
|  |  |
|  | #removes \n from list |
|  | tempList = map(lambda s: s.strip(), tempList) |
|  |  |
|  | #print(tempList) |
|  | #puts items in linked list into list |
|  | from collections import Counter |
|  | c = Counter(tempList) |
|  | print() |
|  | print("20 most common passwords are: ") |
|  | print(c.most\_common(20)) |
|  |  |
|  |  |
|  | #finds length of linked list |
|  | def lLength(llist): |
|  | count = 0 |
|  | while llist is not None: |
|  | count += 1 |
|  | llist = llist.next |
|  | return count |
|  | #sorts linked list in descending order |
|  | def mergeSort(llist): |
|  | if llist == None or llist.next == None: |
|  | return llist |
|  | length = lLength(llist) |
|  | pivot = length//2 |
|  | temp = llist |
|  | for i in range(pivot - 1): |
|  | temp = temp.next |
|  |  |
|  | llist2 = temp.next |
|  | temp.next = None |
|  |  |
|  | sList1 = mergeSort(llist) |
|  |  |
|  | sList2 = mergeSort(llist2) |
|  |  |
|  | if(sList1.password > sList2.password): |
|  | mList = sList1 |
|  | sList1 = sList1.next |
|  | else: |
|  | mList = sList2 |
|  | sList2 = sList2.next |
|  | temp = mList |
|  | while sList1 is not None and sList2 is not None: |
|  | if(sList1.password > sList2.password): |
|  | temp.next = sList1 |
|  | sList1 = sList1.next |
|  | else: |
|  | temp.next = sList2 |
|  | sList2 = sList2.next |
|  | temp = temp.next |
|  |  |
|  | while sList1 is not None: |
|  | temp.next = sList1 |
|  | sList1 = sList1.next |
|  | temp = temp.next |
|  |  |
|  | while sList2 is not None: |
|  | temp.next = sList2 |
|  | sList2 = sList2.next |
|  | temp = temp.next |
|  |  |
|  | return mList |
|  |  |
|  | temp = None |
|  | #opens file and inserts passwords in linked list |
|  | with open('10-million-combos.txt') as f: |
|  | for line in f: |
|  | info = line.split('\t') |
|  | password = info[1] |
|  | # temp = Node(password, 1, temp) |
|  | if not inList(temp, password): |
|  | temp = Node(password, 1, temp) |
|  |  |
|  | root = temp |
|  | mList = mergeSort(root) |
|  | #Prints linked list in descending order |
|  | #printList(mList) |
|  | #Prints 20 most common passwords |
|  | popular(mList) |

“I certify that this project is entirely my own work. I wrote, debugged, and tested the code being presented, performed the experiments, and wrote the report. I also certify that I did not share my code or report or provided inappropriate assistance to any student in the class.”