



Escuela Politécnica Nacional
Facultad de Ingeniería de Sistemas

Ingeniería en Computación

Métodos Numéricos

Ing. Jonathan A. Zea

Integrantes: Roberth Gancino

Danny Iñaguazo

Anthony Goyes

Fecha: 26/06/2024

[Taller 01] Programación

Problema 1. Xplore

El problema a resolver mediante un algoritmo diseñado en Python consiste en obtener el nombre y el h-index de los autores que están citados en N artículos, los cuales son ingresados en formato JSON. Dicho algoritmo debe abarcar dos a diez mil artículos como parámetros de entrada.

Algoritmo

```
1  from collections import OrderedDict
2
3  def funcion(n, data):
4
5      if n == len(data) and n != 0:
6          x = {} #Diccionario que almacena el nombre del autor y un array de sus citaciones
7          if n >= 2 and n <= 10000:
8              for i in range(len(data)):
9                  names = [author["full_name"] for author in data[i]["authors"]["authors"]]
10                 count = data[i]["citing_paper_count"]
11
12                 for aux in range(len(names)):
13                     if names[aux] not in x:
14                         x[names[aux]] = [count]
15                     else:
16                         for aux1 in x.keys():
17                             if aux1 == names[aux]:
18                                 x[aux1].append(count)
19
20                 for key, value in x.items():
21                     sorted_array = sorted(value, reverse=True)
22                     x[key] = sorted_array
23                     j = 0
24                     cont = 0
25                     for j in range(len(sorted_array)):
26                         if sorted_array[j] >= j+1:
27                             cont += 1
28                     x[key].append(cont) #El valor final del array es igual al h-index
29
30                 sorted_data = dict(sorted(x.items(), key=lambda item: (-item[1][-1], item[0])))
31
```

```

32         for f in sorted_data:
33             print(f"{f} {sorted_data[f][len(sorted_data[f])-1]}")
34
35
36
37     elif n == 1:
38         names = [author["full_name"] for author in data[0]["authors"][0]["authors"]]
39         for aux in range(len(names)):
40             x[names[aux]] = 1
41
42         orden = OrderedDict(sorted(x.items()))
43         print(orden)
44         for aux1 in orden:
45             print(f"{aux1} {orden[aux1]}")
46     else:
47         print("*****ERROR*****")
48
49     data = []
50     data.append({"authors": {"authors": [{"author_order": 1,"affiliation": "", "full_name": "Echo"}, {"author_order": 2,"affiliation": "", "full_name": "Charlie"}, {"author_order": 3,"affiliation": "", "full_name": "Echo"}]},"title": "Xplore"})
51     data.append({"authors": {"authors": [{"author_order": 1,"affiliation": "", "full_name": "Echo"}, {"author_order": 2,"affiliation": "", "full_name": "Charlie"}]},"title": "Xplore"})
52     data.append({"authors": {"authors": [{"author_order": 1,"affiliation": "", "full_name": "Charlie"}]},"title": "Xplore"})
53     data.append({"authors": {"authors": [{"author_order": 1,"affiliation": "", "full_name": "Charlie"}, {"author_order": 2,"affiliation": "", "full_name": "Delta"}]},"title": "Xplore"})
54     data.append({"authors": {"authors": [{"author_order": 1,"affiliation": "", "full_name": "Charlie"}]},"title": "Xplore"})
55     data.append({"authors": {"authors": [{"author_order": 1,"affiliation": "", "full_name": "Delta"}]},"title": "Xplore"})
56     data.append({"authors": {"authors": [{"author_order": 1,"affiliation": "", "full_name": "Charlie"}]},"title": "Xplore"})
57     data.append({"authors": {"authors": [{"author_order": 1,"affiliation": "", "full_name": "Delta"}, {"author_order": 2,"affiliation": "", "full_name": "Bravo"}]},"title": "Xplore"})
58     data.append({"authors": {"authors": [{"author_order": 1,"affiliation": "", "full_name": "Bravo"}]},"title": "Xplore"})
59     data.append({"authors": {"authors": [{"author_order": 1,"affiliation": "", "full_name": "Bravo"}]},"title": "Xplore"})
60
61     funcion(10, data)

```

Ejecución del algoritmo

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\USEER\Downloads\Python_Projects> & "C:/Program Files/Python312/python.exe" c:/Users/USEER/Downloads/Python_Projects/Xplore.py
Charlie 5
Echo 4
Alfa 3
Bravo 3
Delta 3
PS C:\Users\USEER\Downloads\Python_Projects>

```

Para la ejecución del algoritmo se utilizaron los mismos parámetros del ejemplo proveído por el problema. De modo que, se puede apreciar que el resultado es el mismo y por ende el algoritmo funciona correctamente.

Csacademy

The image displays two screenshots of the Cscademy task 'Xplore' interface, showing the problem statement and the user's submission results.

Top Screenshot (Statement Tab):

- Standard input:** The input will consist of an integer N , followed by N lines with a single article entry in each line in a JSON format.
- Standard output:** Print the authors ranked by their h -index followed by a space and by the h -index itself. The authors should be ranked alphabetically if there are ties.
- Constraints and notes:** $2 \leq N \leq 10000$
- Input:** A JSON array of N entries, each containing an 'authors' field with an array of author objects. Each author object has 'author_order', 'affiliation', and 'full_name' fields.
- Python Code:** The user's solution is a Python function that processes the input data, calculates the h -index for each author, and prints the authors ranked by their h -index, with ties broken alphabetically.

Bottom Screenshot (Submissions Tab):

- Submissions:** A list of five submissions, all marked as '0 points'.
- Output:** The user's submission shows the following output:

```
1 Charlie 5
2 Echo 4
3 Alfa 3
4 Bravo 3
5 Delta 3
6
```

Se puede apreciar que el algoritmo funciona correctamente, pero a pesar de eso se obtiene una calificación de cero puntos y al subirlo para su calificación siempre marca como si se dieran respuestas incorrectas, a pesar de esto se sabe que dicho algoritmo funciona correctamente, pero algo ocurre al subirlo a Cscademy. Dichos errores podrían deberse a que el algoritmo implementado no es totalmente optimo o talvez la página web esperaba otro tipo de algoritmo.