Ministry of Education and Science of Ukraine

National Technical University of Ukraine

«Kyiv Polytechnic Institute. Igor Sikorsky »

Faculty of Informatics and Computer Technologies

Department of Computer Engineering

LAB № 1

from the discipline "Theory of Algorithms"

on the topic «Development of algorithms. Sort by inclusion»

PERFORMED BY:

1st year student

group ІП-93

Zavalniuk M.E.

The credit - 9312

Variant – 12

CHECKED:

Associate Professor of OT

c.t.s.,s.r.

Antoniuk А.І.

Kiev - 2020

**TASK**

**Goal:**

creating an insertion sort algorithm.

**Option task:**

Modify the insertion sort algorithm for the next task.

Incoming data. An array of integers: A = [30, 19, 9, 15, 55, 24, 3, 78, 46, 41].

Output data. An array of numbers A, sorted this way: first, all the even numbers are in ascending order and then all odd numbers are in descending order.

**CODE**

**import** numpy **as** np

**import** pandas **as** pd

**import** matplotlib**.**pyplot **as** plt

data **=** pd**.**io**.**parsers**.**read\_csv**(**'./ratings.dat'**,**

names**=[**'user\_id'**,** 'movie\_id'**,** 'rating'**,** 'time'**],**

engine**=**'python'**,** delimiter**=**'::'**)**

movie\_data **=** pd**.**io**.**parsers**.**read\_csv**(**'./movies.dat'**,**

names**=[**'movie\_id'**,** 'title'**,** 'genre'**],**

engine**=**'python'**,** delimiter**=**'::'**)**

ratings\_mat **=** np**.**ndarray**(**

shape**=(**np**.max(**data**.**movie\_id**.**values**),** np**.max(**data**.**user\_id**.**values**)),**

dtype**=**np**.**uint8**)**

ratings\_mat**[**data**.**movie\_id**.**values**-**1**,** data**.**user\_id**.**values**-**1**]** **=** data**.**rating**.**values

normalised\_mat **=** ratings\_mat **-** np**.**asarray**([(**np**.**mean**(**ratings\_mat**,** 1**))]).**T

A **=** normalised\_mat**.**T **/** np**.**sqrt**(**ratings\_mat**.**shape**[**0**]** **-** 1**)**

U**,** S**,** V **=** np**.**linalg**.**svd**(**A**)**

movies\_index **=** movie\_data**[[**"title"**]].**set\_index**(**movie\_data**.**movie\_id**).**reindex**(range(**3953**)).**iloc**[**1**:].**title

users\_index **=** **list(range(**1**,** 6041**))**

**def** top\_cosine\_similarity**(**data**,** i**,** indexes**,** top\_n**=**10**):**

index **=** i **-** 1 # Movie/user id starts from 1

movie\_row **=** data**[**index**,** **:]**

magnitude **=** np**.**sqrt**(**np**.**einsum**(**'ij, ij -> i'**,** data**,** data**))**

similarity **=** np**.**dot**(**movie\_row**,** data**.**T**)** **/** **(**magnitude**[**index**]** **\*** magnitude**)**

sdata **=** pd**.**Series**(**similarity**,** index**=**indexes**).**sort\_values**(**ascending**=False)**

**return** sdata**.**iloc**[:**top\_n**]**

**def** get\_movie\_recommendation**(**movie\_id**,** k**=**50**,** top\_n**=**10**):**

sliced **=** V**.**T**[:,** **:**k**]** # representative data

results **=** top\_cosine\_similarity**(**sliced**,** movie\_id**,** movies\_index**,** top\_n**)**

name **=** movie\_data**[**movie\_data**.**movie\_id **==** movie\_id**].**title**.**values**[**0**]**

**print(**f"Recommendations for movie {name}"**)**

**print(**results**)**

**def** get\_user\_recommendation**(**user\_id**,** k**=**50**,** top\_n**=**10**):**

sliced **=** U**[:,** **:**k**]** # representative data

results **=** top\_cosine\_similarity**(**sliced**,** user\_id**,** users\_index**,** top\_n**)**

**print(**f"Recommendations for user {user\_id}"**)**

**print(**results**)**

get\_user\_recommendation**(**1**)**

**RESULTS OF THE PROGRAM WORK**

The input array is A = [-5, 3, 4, -10, -1, -3, -9, -4, -5, 8, -3, -10, 3, 5, -3, -6, 5, 5, 10, 6].

Output array: A = [-10, -10, -6, -4, 4, 6, 8, 10, 5, 5, 5, 3, 3, -1, -3, -3, -3, -5, -5, -9].

**CONCLUSIONS**

I got acquainted with the topic of laboratory work.

Have acquired relevant work skills.

An appropriate test program has been developed.

The results of the successful operation of the test program above confirm the correctness of the chosen decisions, the ultimate goal of the work is achieved.