ISE4 Mini Project

Group 19

Book Scanning

Team Members:

- 1. Aditi Bhosale (2019140009)
- 2. Anushka Agarwal (2019140002)
- 3. Shweta Chavan (2019140011)

Project Code:

```
import numpy as np
import operator
import sys
# signup score and library. We have to sort based on signup score, and hence item
getter(0) since based on array1 we are sorting
def parallel sort(array 1, array 2, ascending=False):
    if not ascending:
        return zip(*sorted(zip(array 1, array 2), key=operator.itemgetter(0))[::-1])
    return zip(*sorted(zip(array_1, array_2), key=operator.itemgetter(0)))
# zip is to create two arrays such that their keys are matching.
class Library:
   def _ init (self, id, book ids, signup days, max books_scanned_per_day):
        self.id = id
        self.book ids = book ids
        self.signup days = signup days
        self.max books scanned per day = max books scanned per day
    def get_best_book_ids(self, start_day=0):
        global book scores, libraries signup days,
libraries max books scanned per day, D, final books
        available_days = D - self.signup_days - start_day #start day is different for different libraries, and available days also
depends on previous libraries which had been there before.
        available books = list(set(self.book ids) - set(final books))
        current book scores = np.take(book scores, available books)
        # returns array of same type with all the values of given indices. So here
        # book scores of all available books
```

```
max num books = max(min(int(available days * self.max books scanned per day),
len(available books)), 0)
       # if available books is 1 and max scan possible is 2 then we have to scan 1
       if max num books == 0:
           return []
       ind = np.argpartition(current book scores, -max num books)[-max num books:]
       # argpartition returns the array with index of the books in sorted order, and
we have passed all values here
       return np.take(available books, ind)
   def get best books score(self, start day=0):
       global book scores, libraries signup days,
libraries max books scanned per day, D
       available days = D - self.signup days - start day
       max num books = max(min(int(available days * self.max books scanned per day),
len(self.book ids)), 0)
       current book scores = np.take(book scores, self.book ids)
       # get top k books (k=max num books) argpartition returns the sorted index of
all books since all values till max books #are passed one by one
       ind = np.argpartition(current book scores, -max num books)[-max num books:]
       best books scores = np.take(current book scores, ind)
       return np.sum(best books scores)
    # returns string representation of object
def sum book scores(book ids):
   global book scores
    return np.sum(np.take(book scores, list(book ids)))
files = [
total score = 0
for file in files:
   with open("inputs/" + file + ".txt", "r") as f:
       content = f.read().splitlines() # read each input lines differently
   print(file) # prints the file name
   for z in range(60):
   book scores = list(map(int, content[1].split(' ')))
```

```
for x in range(len(book scores)):
       print(book scores[x], end=" ")
    # library attributes : hence size of all below parameters are L
    libraries num books = np.zeros(L) # number of books in lib 0,1,2 etc hence array
   libraries signup days = np.zeros(L) # signup days for the libraries hence array
    libraries max books scanned per day = np.zeros(L) # max books canned per day for
    libraries = np.empty(L, dtype=Library) # array number of elements in row is L
and data type is library
   for i in range(L):
       pos += 1
       n, t, m = list(map(int, content[pos].split(' ')))
       libraries num books[i] = n
       libraries_signup_days[i] = t
       libraries_max_books_scanned_per_day[i] = m
       pos += 1
       book ids = np.asarray(list(map(int, content[pos].split(' '))))
       for x in range(len(book ids)):
           print("\t", book ids[x], end=" ")
       libraries[i] = Library(i, book ids, t, m)
    # Vectorization is used to speed up the Python code without using loop.
    library book score counter = np.vectorize(lambda library:
library.get best books score())
    libraries_scores = library_book_score_counter(libraries)
# we pass the array libraries and vectorize takes in each element of the array passed
and gets the best book score
   heuristic score = np.vectorize(lambda book score, signup days: book score /
signup days)
    signup scores = heuristic score(libraries scores, libraries signup days)
    signup scores, libraries sorted = parallel sort(signup scores, libraries)
    final books = set()
            "outputs/" + file + ".out", 'w+') as f:
       print()
        for z in range(60):
    str(L))
```

```
start_day = 0
        for i in range(L):
            current_library = libraries_sorted[i]
            chosen_book_ids = current_library.get_best_book_ids(start_day)
final_books.update(chosen_book_ids)
            start_day += current_library.signup_days
            if len(chosen book ids) > 0:
                 f.write(str(current_library.id) + " " + str(len(chosen book ids)) +
                 f.write(str(' '.join(map(str, chosen_book_ids))) + "\n")
                 f.write(str(current_library.id) + " 1\n")
                 f.write(str(current library.book ids[0]) + "\n")
            if len(chosen_book_ids) > 0:
                       str(current library.id) + " \t\t\t\t\t\t\t\t\t\t\t\t" +
str(len(chosen book ids)) + "\n")
                print(str(' '.join(map(str, chosen_book_ids))))
                print(str(current library.id) + " 1\n")
                print(str(current_library.book ids[0]) + "\n")
             # progress = 100 * i / (2 * L)
             # sys.stdout.write("\rCreating output... (" + str(int(progress)) + " %)")
        score = sum book scores(final books)
print("")
print("\n\t\t\t\t\t\t\t\t\t\t\t\t\tTotal score:", total score, "\n\n")
for z in range(60):
```

Project Output:

Test Case 1:

```
a_example
Enter the number of Books, Libraries, Days:
                       6 2 7
                       Enter the Book scores: 1 2 3 6 5 4
                        -******* Library 0 *******
                       Books Signup days Books shipped per day
                       Enter Book ids: 0 1 2 3 4
                        -******* Library 1 ******
                       Books Signup days Books shipped per day
                       Enter Book ids: 0 2 3 5
Number of libraries scanned up for signing = 2
                       Library chosen
                                       Number of books scanned from library
                           0
                       Book ID's of books chosen from library : 0 1 2 3 4
Number of libraries scanned up for signing = 2
                      Library chosen
                                 Number of books scanned from library
                      Book ID's of books chosen from library : 0 1 2 3 4 \,
                      Library chosen
                                      Number of books scanned from library
                      Book ID's of books chosen from library : 5
                      Total score: 21
Process finished with exit code 0
```

Test Case 2:

b_read_on

Total score: 5822900

Process finished with exit code 0

Test Case 3:

c_incunabula

Total score: 5645747

Process finished with exit code 0

Test Case 4:

d_tough_choices

Total score: 4812730

Process finished with exit code 0

Test Case 5:

e_so_many_books

Total score: 5019670

Process finished with exit code 0

Test Case 6:

f_libraries_of_the_world

Total score: 5240161

Process finished with exit code 0