
PPPD - Lab. 11

Copyright ©2022 M. Śleszyńska-Nowak i in.

Zadanie punktowane, lab 11, grupa B, 2022/2023, autor: Małgorzata Śleszyńska-Nowak

Uwaga: w rozwiązaniu zadania nie można używać slice'ów ani indeksowania ujemnego.

Temat: Right-join i najlepszy zespół

Treść zadania

Zadanie polega na implementacji sposobu łączenia tabel danych znanego jako RIGHT-JOIN (złączenie zewnętrzne prawostronne). W zadaniu zakładamy, że mamy dwie macierze M1 i M2, każda z nich składa się z dowolnej liczby wierszy i kolumn, wiersze o indeksie 0 zawierają nagłówki kolumn, kolumny o indeksie 0 to ID, po którym będziemy dokonywać złączenia.

Right-join macierzy M1 i M2 polega na dołączeniu do macierzy M1 rekordów z macierzy M2 dla tych przypadków, w których wartości w kolumnie ID są sobie równe. Jeśli w macierzy M1 istnieje ID, którego nie ma w macierzy M2, rekord ten jest pomijany. Jeśli w macierzy M2 istnieje ID, którego nie ma w macierzy M1, dodajemy wiersz z tym ID, a we wszystkich kolumnach odpowiadających wartościom z macierzy M1 wstawiamy *NaN* (`float("NaN")`). Kolejność wierszy w wynikowej macierzy jest następująca: najpierw wiersze z ID, które jest w obu macierzach, w kolejności jak w macierzy M1, następnie wiersze z ID występującym tylko w M2, w kolejności jak w macierzy M2.

Przykład:

M1:

ID	M11	M12
5	0	"a"
1	1	"b"
3	2	"d"
2	3	"a"
7	4	"t"

M2:

ID	M21	M22
8	"x"	1
0	"y"	5
2	"z"	9
3	"a"	0
5	"o"	1

`right_join(M1, M2):`

ID	M11	M12	M21	M22
5	0	"a"	"o"	1
3	2	"d"	"a"	0
2	3	"a"	"z"	9
8	NaN	NaN	"x"	1
0	NaN	NaN	"y"	5

W zadaniu użyjemy `right-join` w pracy z danymi pracowników pewnej firmy. Wczytamy dwie macierze: $M1$ z informacjami o przynależności pracowników do zespołów (dwie kolumny: `id_pracownika` i `id_zespołu`) i $M2$ z podsumowaniem wyników sprzedawców z ostatniego tygodnia (kolumny: `id_pracownika` i pewna liczba kolumn odpowiadających różnym rodzajom zadań wykonywanych przez sprzedawców). Po odpowiednim połączeniu macierzy dokonamy podsumowania wyników wszystkich sprzedawców i znajdziemy najlepszy zespół.

Do wczytania danych z plików użyj funkcji:

```
import csv
def read_csv(plik):
    data_frame = []
    with open(plik) as f:
        csv_reader = csv.reader(f)
        headings_row = next(csv_reader)
        data_frame.append([headings_row[i] for i in range(len(headings_row))])
        for row in csv_reader:
            data_frame.append([int(row[i]) for i in range(len(row))])
    return data_frame
```

Wszystkie wiersze oprócz zerowego zawierać będą wartości całkowite (`int`).

Do wypisywania macierzy użyj funkcji:

```
def print_data(M):
    for i in range(len(M)):
        for j in range(len(M[i])):
            print(f"{M[i][j]:^10}", end=" ")
        print()
```

Napisz program składający się z funkcji:

- `right_join(M1, M2)` - przyjmuje macierze $M1$ i $M2$, zwraca macierz powstałą poprzez złączenie zewnętrzne prawostronne macierzy $M1$ i $M2$ zgodnie z opisem podanym powyżej. Funkcja ma działać dla dowolnych macierzy (o dowolnej liczbie kolumn i wierszy), z założeniem, że łączymy po ID, które znajduje się w kolumnach o indeksie 0. Funkcja nie może modyfikować przyjętych argumentów, typy wszystkich rekordów pozostają takie same jak w przyjętych macierzach.
- `evaluate(M, r)` - przyjmuje macierz M , liczbę całkowitą $team$, oraz listę liczb całkowitych r
 - M to macierz z danymi dotyczącymi sprzedawców, w której zerowy wiersz to wiersz z nagłówkami, a każdy następny wiersz odpowiada jednemu sprzedawcy i zawiera pod indeksem 0: ID pracownika, pod indeksem 1: ID zespołu, a w kolejnych komórkach liczbę zrealizowanych zadań różnych typów
 - $team$ to ID konkretnego zespołu
 - r to lista liczb całkowitych o długości równej liczbie różnych zadań realizowanych przez sprzedawców - każdy element mówi ile punktów sprzedawca dostaje za zrealizowanie jednego zadania odpowiedniego typu
 - funkcja zwraca macierz z podsumowaniem punktów sprzedawców należących do zespołu $team$. W wynikowej macierzy każdy wiersz składa się z dwóch kolumn: ID pracownika, suma punktów zdobytych przez pracownika
 - funkcja nie może modyfikować przyjętych argumentów
- `best_team(M, teams, r)` - przyjmuje macierz M oraz dwie listy liczb całkowitych: $teams$ i r
 - M i r analogiczne jak w funkcji `evaluate(M, r)`
 - $teams$ to lista zawierająca ID zespołów
 - funkcja zwraca krotkę: ID najlepszego zespołu, jego średnia liczba punktów. Najlepszy zespół to zespół, który uzyskał najlepszy średni wynik (tzn. sumę punktów pracowników z danego zespołu podzieloną przez liczbę tych pracowników), wskazówka: użyj napisanej wcześniej `evaluate(M, r)`

-
- funkcja nie może modyfikować przyjętych argumentów
 - `main` - wywołuje wszystkie funkcje:
 - wczytuje dane z pliku `m1B.csv` do macierzy `M1` (przypisanie pracowników do zespołów), dane z pliku `m2B.csv` do macierzy `M2` (wyniki pracy sprzedawców)
 - wywołuje `right_join` na macierzach `M1` i `M2` (W `M1` mamy wszystkich pracowników, którzy są przypisani do jakichś zespołów, w `M2` mamy sprzedawców - jeżeli pracownik nie jest sprzedawcą to nie rozważamy go w podsumowaniu wyników - dlatego potrzebujemy `right-joina`)
 - wypisuje macierz zwróconą przez `right_join`
 - dla `r = [1, 1.5, 2, -0.5]` oraz `teams = [100, 101, 103, 104, 105]` znajduje najlepszy zespół i wypisuje stosowne informacje

Punktacja

Za poszczególne elementy można uzyskać następującą liczbę punktów:

- `right_join` - 5 punktów
- `evaluate` - 2 punkty
- `best_team` - 2 punkty
- `main` - 1 punkt

Uwagi

- Jeśli rozwiązanie nie spełnia postawionych wymagań (korzysta ze `slice'ów` lub indeksowania ujemnego), zadanie jest oceniane na 0 punktów.
- Jeśli program się nie kompiluje (interpretuje), ocena jest zmniejszana o połowę
- Jeśli kod programu jest niskiej jakości (nieestetycznie formatowanie, mylące nazwy zmiennych itp.), ocena jest zmniejszana o 2p

Przykłady interakcji użytkownika z programem

Dla danych z plików `m1B.csv` i `m2B.csv`

Po `right-join`

<code>id_prac</code>	<code>id_zesp</code>	<code>male_tr</code>	<code>srednie_tr</code>	<code>duze_tr</code>	<code>skargi</code>
198700	100	10	5	3	0
198701	103	17	4	0	0
198702	100	17	3	1	1
198900	104	3	5	10	0
198901	103	9	10	9	0
199500	104	15	10	5	5
199501	105	12	6	3	0
199602	105	16	4	2	1
199900	103	10	7	3	0
200000	101	20	0	0	1
200001	100	10	10	5	2
200005	nan	11	9	5	1
199603	nan	18	3	2	0
198706	nan	10	10	10	1
198904	nan	8	7	11	0

Zespół 104 uzyskał najwyższą średnią ocenę: 34.0

Process finished with exit code 0