[SOLUTION] CS-465-204-AI-Lab-Finalterm-Version[A]

April 19, 2023

1 Finalterm Exam

1.0.1 Question # 01

Write a Python function that takes in a NumPy array of integers and returns a new array where each element is replaced by the product of all the other elements in the original array. For example, given the following input array:

`arr = np.array([1, 2, 3, 4])`

The function should return the following output array: `array([24, 12, 8, 6]`, the first element of the output array is the product of all the other elements in the input array (2 * 3 * 4 = 24), the second element is the product of all the other elements except the second one (1 * 3 * 4 = 12), and so on.

[1]:
```
import numpy as np
def product_of_others(arr):
    product = np.prod(arr)
    return np.array([product // x for x in arr])

arr = np.array([1, 2, 3, 4])
result = product_of_others(arr)
print(result)
```

```
[24 12  8  6]
```

1.0.2 Question # 02

By Using pandas library, Answer the following questions from "Automobile_data.csv"

```
i.   Find the most expensive car company name
ii.  Count total cars per company
iii. Print All Toyota Cars details
iv.  Find the average mileage of each car making company
```

[2]:
```
import pandas as pd
data = pd.read_csv('Automobile_data.csv')
df = pd.DataFrame(data)
```

[3]:
```
df[['company', 'price']][df['price'] == df['price'].max()]
```

[3]:
```
           company    price
35  mercedes-benz  45400.0
```

[4]:
```
df['company'].value_counts()
```

[4]:
```
toyota           7
bmw              6
mazda            5
nissan           5
audi             4
mercedes-benz    4
mitsubishi       4
volkswagen       4
alfa-romero      3
chevrolet        3
honda            3
isuzu            3
jaguar           3
porsche          3
dodge            2
volvo            2
Name: company, dtype: int64
```

[5]:
```
df[df['company'] == 'toyota']
```

[5]:

	index	company	body-style	wheel-base	length	engine-type	num-of-cylinders	horsepower	average-mileage	price
48	66	toyota	hatchback	95.7	158.7	ohc	four	62	35	5348.0
49	67	toyota	hatchback	95.7	158.7	ohc	four	62	31	6338.0
50	68	toyota	hatchback	95.7	158.7	ohc	four	62	31	6488.0
51	69	toyota	wagon	95.7	169.7	ohc	four	62	31	6918.0
52	70	toyota	wagon	95.7	169.7	ohc	four	62	27	7898.0
53	71	toyota	wagon	95.7	169.7	ohc	four	62	27	8778.0
54	79	toyota	wagon	104.5	187.8	dohc	six	156	19	15750.0

[6]:
```
companies = df.groupby('company')
companies[['average-mileage']].mean()
```

[6]:

company	average-mileage
alfa-romero	20.333333
audi	20.000000
bmw	19.000000
chevrolet	41.000000
dodge	31.000000
honda	26.333333
isuzu	33.333333
jaguar	14.333333
mazda	28.000000
mercedes-benz	18.000000
mitsubishi	29.500000
nissan	31.400000
porsche	17.000000
toyota	28.714286
volkswagen	31.750000
volvo	23.000000

1.0.3 Question # 03

You have been hired by GIFT University to assist in the management of the timetable of the classes. Your first task is managing the timetable of the CS department's new Data Science BS program. The classes need to be held weekly on three days, that is Mondays, Wednesdays, and Fridays only. There would be a total of 8 lectures to be held on these days and a total of 5 newly hired Ph.D. teachers would be teaching these classes. You are limited by the fact that each teacher can only teach one class at a time.

The lectures to be scheduled are:

- `Lecture 01 - Data Science Fundamentals: 8:00AM - 9:00AM`
- `Lecture 02 - Programming for Data Science: 8:30AM - 9:30AM`
- `Lecture 03 - Data Analysis: 9:00AM - 10:00AM`
- `Lecture 04 - Fundamentals of AI: 9:00AM - 10:00AM`
- `Lecture 05 - Problem Solving and Machine Learning: 9:30AM - 10:30AM`
- `Lecture 06 - Machine Learning: 10:00AM - 11:00AM`
- `Lecture 07 - Statistics & Probabilities: 10:30AM - 11:30AM`
- `Lecture 08 - Deep Learning: 11:00AM - 12:00PM`

The teachers who would be taking these lectures and their lecture preferences would be:

- `Ahmed, who would be teaching Lectures 3, 4 and 5.`
- `Osama, who would be teaching Lectures 4, 6 and 7.`
- `Farhan, who would be teaching Lectures 5, 6, 7 and 8.`
- `Uzair, who would be teaching Lectures 2, 3, 4, and 5,`
- `Sadia, who would be teaching Lectures 1, 2, 3, 4, 5.`

Using backtrack, solve this problem as a CSP problem in which there is one variable per lecture, stating the domains, and constraints. Constraints should be specified formally and precisely.

[7]:
```python
from simpleai.search import CspProblem, backtrack

variable = ('L1', 'L2', 'L3', 'L4', 'L5', 'L6', 'L7', 'L8')

def const(variable, values):
    return values[0] != values[1]

domain = {
    'L1': ['Sadia'],
    'L2': ['Uzair', 'Sadia'],
    'L3': ['Ahmed', 'Uzair', 'Sadia'],
    'L4': ['Ahmed', 'Osama', 'Uzair', 'Sadia'],
    'L5': ['Ahmed', 'Farhan', 'Uzair', 'Sadia'],
    'L6': ['Osama', 'Farhan'],
    'L7': ['Osama', 'Farhan'],
    'L8': ['Farhan'],
}

constraint = [
    (('L1', 'L2'), const),
    (('L2', 'L3'), const),
    (('L2', 'L4'), const),
    (('L3', 'L4'), const),
    (('L3', 'L5'), const),
    (('L4', 'L5'), const),
    (('L5', 'L6'), const),
    (('L6', 'L7'), const),
    (('L7', 'L8'), const),
]

problem = CspProblem(variable, domain, constraint)
output = backtrack(problem)
for k, v in output.items():
    print(k, '==>', v)
```

```
L1 ==> Sadia
L2 ==> Uzair
L3 ==> Ahmed
L4 ==> Osama
L5 ==> Uzair
L6 ==> Farhan
L7 ==> Osama
L8 ==> Farhan
```