python_exam_programming

June 7, 2020

1 Least cost path problem

Maximum marks : 10

- You are given an array `costs` of shape (M,N) of positive integers.
- Your goal is to find the least cost path from the lower left corner (M-1,0) to top right corner (0,N-1). Let's call these places START and FINISH respectively.
- You can only move UP or RIGHT. A valid path is thus a sequence made of {UP,RIGHT} values, which takes you from START to FINISH.
- The cost of the path is the sum of the elements in `costs` through which the path must pass.

Consider the following array as `costs`

5	7	5	4	1
6	1	2	3	2
4	3	5	5	4
1	2	3	4	5

One example of a valid path is

5	7	5	4	**1**
6	1	**2**	**3**	**2**
4	**3**	**5**	5	4
1	**2**	3	4	5

Its cost is `1 + 2 + 3 + 5 + 2 + 3 + 2 + 1` = `19`.

You can easily verify that the least cost path is

5	7	5	4	**1**
6	**1**	**2**	**3**	**2**
4	**3**	5	5	4
1	**2**	3	4	5

Submit a file `least_cost.py` with the following functions. Assume that costs can have arbitrary shape, but is 2D.

- `least_cost(costs)` : Returns the least cost required to travel from `costs[-1,0]` to `costs[0,-1]` (**6 marks**)
- `least_cost_2(costs,i,j)` : Returns the least cost required to travel from `costs[i,0]` to `costs[j,-1]`, where i>=j (so that you can reach your required position only using UP and RIGHT) (**2 marks**)
- `least_cost_3(costs)` : Along with returning the least cost, also returns the coordinates of the least cost path as a list of tuples (**2 marks**).

Hints : 1. You can use recursion for this. This is very similar to the staircase problem discussed on 6 June. How can we define the larger problem in terms of smaller problems? (hint : cost(A,B) = min of ???)

2 Numpy data analysis

Write a script `numpy_exam.py` that performs the following function. You may not use `pandas` or any other library other than `numpy` for the following problems.

For these questions you will require the numpy load and save functions. Please go through the linked documentation if you are not familiar with them.

Maximum marks : 15

Q1. Load the file `data.npy` using the numpy load function. Let this be called `data`. **(1 marks)**

`Information`

- `data` is a 2D float array whose second dimension is 4.
- Each row in `data` represents a unique person.
- The columns correspond to attributes/qualities of that person.
- They are age(yrs), height(m), weight(kg) and gender (0 for Male, 1 for female)

Q2. Each of the following questions has **2marks** for a total of **10 marks**.

1. Sort the array according to age, and save it as `data_age_sorted.npy`.
2. Sort the male rows by height and save it as `data_males_height_sorted.npy`.
3. Print the difference in mean values for age, height and width between males and females.
4. Print how the top 25% tallest people are divided among males and females (in terms of percentages).
5. Find the indices of the people who are in the heaviest 50 people as well as in the tallest 50 people. Save it as `big_people_idcs.npy`

Q3. Print the average difference in heights and weights between any man and woman.

`Information`

- Given any (man, woman) combination, we can compute `height_diff=man_height - woman_height`.
- Thus the goal is to calculate the average of `height_diff` for every (man,woman) combination. Similarly for weight.

You will be scored on the basis of how vectorized your code is,

- **2 marks** for using 2 or more loops.
- **4 marks** for using only 1 loop.

- **6 marks** for using no loops.