15-110: Principles of Computing

Homework 07

Due: 4th October, 2022 at 10:00pm

- You need to complete the Python file for this assignment, and submit it to Gradescope.
- There are 100 points.
- You must solve the tasks **individually**, always abiding by the course and CMU's academic integrity policy.
- We are not giving you any starter code this week. That means you need to create your file from scratch and define your own testcases. For writing testcases, you may follow the style of testcases used in the previous homework.

1. (35 points) **Daisies Field**

   Nour is the manager of a company that grows daisies for distributing to flower shops around the country. The daisies are grown in vases placed on a rectangular field (forming sort of a grid), with R rows and C columns of vases. Since this is a very big field, Nour has split it into $N \times M$ tiles. She made sure that N divides R and M divides C so that all the field is covered by tiles of the same size, and there is no overlap.

   Harvesting is done one tile at a time, collecting all daisies of that tile that have blossomed. Each day, Nour must decide which tile is going to be harvested, and she would like to choose the tile with the maximum number of blossomed daisies. Implement the function `daisiesField(F, n, m)` to help Nour with this task.

   The function takes as input the field of daises `F` as a list of lists. Each position indicates the number of blossomed daisies on that vase. For example, a 3x2 field where all vases have 2 blossomed daisies is represented as: `[[2,2],[2,2],[2,2]]`. The function also takes as input the number of rows n and columns m of each tile.

   Return the maximum number of daisies that can be collected that day.

   For example:

   - `daisiesField([[2,1,3],[1,1,1]], 1, 1) == 3`
   - `daisiesField([[2,1,3,4],[9,4,2,2]], 2, 2) == 16`
   - `daisiesField([[1,2,3,4], [5,6,7,8], [1,10,5,2], [1,5,9,10]], 2, 1) == 15`

2. (30 points) **Manipulate list values**

   Implement the function `listDiscount(L, discount)` that takes a list of numbers, `L`, and a real number `discount` (representing a discount factor), and returns a tuple of two elements: the original list with each element discounted (i.e., multiplied) by `discount` and the median among the values of the original list `L`.

   The median value is the "middle" value of a sorted sequence of values. If the number of values is an even number n, then the returned median should correspond to the value with position index

$\frac{n}{2}$ in the list. For instance, the median of `[1,2,3,4,5]` is 3, and the median of `[1,2,3,4]` is also 3.

Inside `listDiscount` you must use of a helper function `get_median(LL)` that returns the median value of an input list `LL` and doesn't change the input list `LL` (i.e., make a clone of `LL` or use functions that do not make in-place changes).

For instance:

- `listDiscount([1,2,3,4,5], 0.5)` returns `([0.5, 1.0, 1.5, 2.0, 2.5], 3)`
- `listDiscount([1,2,4,5], 0.5)` returns `([0.5, 1.0, 2.0, 2.5], 4)`
- `listDiscount([2,2,3,1], 0.1)` returns `([0.2, 0.2, 0.3, 0.1], 2)`
- `listDiscount([0,0,0,1,2], 1.5)` returns `([0.0, 0.0, 0.0, 1.5, 3.0], 0)`

3. (35 points) **Digits count**

Given two numbers a and b, count how many times each of the digits 0 to 9 occur in all numbers between a and b, inclusive.

In order to make your code simpler, implement first the function `intToList(n)` that takes as input one integer `n`, and returns a list with the digits of `n` in the order they appear. For example, `intToList(1964)` should return `[1,9,6,4]`.

Using the function `intToList`, implement the function `digitsCount(a, b)` that returns a list with 10 numbers, where position i holds the number of times digit i occurred in all numbers between `a` and `b`, inclusive.

For example:

- `digitsCount(1, 9)` returns the list `[0,1,1,1,1,1,1,1,1,1]`
- `digitsCount(10,15)` returns the list `[1,7,1,1,1,1,0,0,0,0]`