Dynamic Programming Project (Fall 2021/2022)

LadyBug & the Miraculous

Ladybug (Marinette) is a French superhero (sounds like an oxymoron, I know, right!) who mainly operates in Paris. She fights a villain called Hawkmoth with the help of her friend Cat Noir (Adrien). Marinette likes Adrian and Cat Noir likes Ladybug, but ladybug sees Cat Noir as a friend and Adrian sees Marinette as a friend. Is this the usual triangle? I am not sure, to me it seems like a rectangle that is pretending to be a triangle which is actually not (you need three points to form a triangle and here you only have two if you look closely!). Anyway, none of this is really related to our problem, so ignore all that.

Ladybug gains her powers by obtaining objects with magical powers that are called "Miraculous", whenever Lady bug wields a "miraculous" she gains a special power and her energy level increases. Unfortunately, the various miraculous objects have been scattered long ago across the world. Each one was hidden in a secret location. Luckily, Ladybug has a map where the location of each miraculous object is identified as a point on a 2D cartesian coordinate plane (Px, Py). The map shows a total of N locations (on each location you can find a miraculous object).

We are going to help Ladybug in her final battle with Hawkmoth, she starts by visiting location number 1 (where she obtains miraculous object number 1), in addition to gaining energy, this miraculous gives her the ability to fly. Now she will fly from location to location, collecting miraculous objects until she reaches the last location (location N) where Hawk Moth is waiting for her.

If she is currently on location M and flies to location N, she will obtain the miraculous object N which will give Ladybug a specific amount of energy levels but since flying is tiring her energy will also drop by the total distance travelled from M to N. This is why she sometimes skips a certain miraculous object. However, she cannot skip the first and last locations/objects

(because the first one will give her power to fly and the last one is where she will find Hawkmoth).

Her chances of winning over Hawkmoth is related to her energy level when she arrives to the last location, so we have to help Ladybug reach Hawkmoth with the maximum possible energy level. You can assume that she starts with zero energy level before visiting the first location.

The Input Format:

- N (integer, number of locations/objects)

- N lines, each line describes a location. For that location the following is specified: Px, Py and Energy. (Px, Py specifies the coordinate of the miraculous object and Energy specifies the amount of increase you gain when taking that object).

The Output Format:

- N (maximum remaining energy when Ladybug arrives at the last location)

The above example would be presented as follows on the hackerrank:

Input:

3

1 2 2

4 2 3

5 5 8

Output:

6.8377

Here is a reminder of the outline of your solution:

Part1: Divide & Conquer

1- Define the value returned by the function *f* which we want to optimize.

2- Define the parameters which *f* depends on.

3- Draw the **recursion tree** for *f* using the values from the example above.

4- Write the recursive (divide and conquer) code to solve the question.

Part2: Dynamic Programming

5- Draw the table and determine the dependencies between the table cells.

6- Determine the direction of movement within the table.

7- Write the Dynamic programming code which fills the table(s).

8- Write the code that will print the final answer.

You are requested to submit a report that explains each of the steps above which also includes graphs and figures to explain your solution and the rationale behind it. Also, you need to submit your working code on the hackerrank website as usual. The link will be provided on Moodle/FB Group.

Important:

To solve this task you are not allowed to copy/be inspired by any piece of code from the internet or from a colleague or from anyone or any place.

If any percentage of resemblance is found between your code and a code listed on the internet (even if the code is a solution for a different problem), it will be considered cheating.

You are only allowed to check the code of the three problems we studied at class and the last years problems I posted on Moodle.

If it has been proven that you cheated on this task (no matter how small the percentage is), you will get zero in the final exam mark.

One final hint to help you with this question: This question is very similar to one of the 3 DP examples we studied in class.

Good Luck

Dr. Samer Arandi