

Forum:[Hack Forums](#)

Thread: [Algorithm mini project](#)

sonic2 - May 17, 2019, 5:27 pm

greetings everyone,

is anyone familiar with dynamic programming, greedy method, recurrence formula or naive recursive algorithm ?

i need help .

Quote:The data of this problem are:

- A series of n aligned locations whose indexes are between 0 and $n-1$. We can only move from left to right.
- A list C of n values which are the amount that we can collect by visiting each location. The first value corresponds to the first location, the second value to the second location, and so on. All these values are positive or null.
- A list A of n values which are the symbol of each location. The first value corresponds to the first location, the second value to the second location, and so on. These values can be positive or negative.
- Two integers A and B positive or negative. During our path, necessarily from left to right, we can choose to visit or not a location.

If we don't visit a location, we collect an amount equal to 0.

If we visit the location of index i , we collect an amount equal to:

- $C[i]$ if this location has the same symbol $A[i]$ as the previously visited location.
- $C[i]$ if this location is the first visited or if it has a symbol $A[i]$ different from the previously visited location.

The goal of this problem is to find out the maximum amount that can be collected during a path.

Example 1: we consider $T = [9, 7, 8, 7, 10, 7]$, $C = [2, 1, 1, 4, 4, 2]$, $A = -2$ and $B = 5$. The maximum amount that can be collected is equal to 170 and corresponds to the visit of locations of indexes 0, 2, 4 and 5.

Example 2: we consider $T = [3, 9, 2, 7, 3, 1]$, $C = [2, 2, 5, 4, 2, 1]$, $A = 2$ and $B = -5$. The maximum amount that can be collected is equal to 9 and corresponds to the visit of locations of indexes 0, 1 and 4.

GREEDY METHOD

Quote:

Propose and implement in Python a greedy strategy to determine the path.

This function will return the amount collected during the path and a list of the indexes of the visited locations.

Is this method optimal? Justify your answer by giving a proof or a counterexample depending on your answer.

RECURRENCE FORMULA AND NAIVE RECURSIVE ALGORITHM

Quote:A RECURRENCE FORMULA

Let n be the number of locations. For $0 \leq i \leq n-1$ let $M[i]$ be the maximum amount that can be collected between the locations of indexes i and $n-1$ (both included).

Find out a recurrence formula for computing $M[i]$. Justify precisely your reasoning.

3.2 A NAIVE RECURSIVE ALGORITHM

Use the previous recurrence formula to implement in Python a naive recursive algorithm to compute the maximum amount collected during a path. Compute the complexity of this algorithm.

DYNAMIC PROGRAMMING

Quote:TOP DOWN

Improve the algorithm of the subpart 3.2 with a Top Down dynamic programming method.

Compute the complexity of this algorithm.

4.2 BOTTOM UP

Implement in Python a Bottom Up dynamic programming algorithm to compute the maximum amount collected during a path.

Complete this algorithm in order to obtain the list of the indexes of the visited locations.

Compute the complexity of this algorithm.

APPLICATIONS

Quote:Apply the previous algorithms to the values contained in the file "MP.txt" attached to this subject. The elements of the first column are the values of A and the elements of the second column are the values of B . We will take $A=-3$ and $B=7$.

samsepi01 - May 17, 2019, 6:11 pm

No, we aren't going to do your homework for you. At least make an effort, post some code you wrote that isn't working or something.

Kay - May 17, 2019, 7:43 pm

There are certainly plenty of implementations out there for that you are trying to accomplish. I would suggest starting by googling some of those key terms, as well as looking on stack overflow and git repos. If you still need help after that, send me a PM.

sonic2 - May 18, 2019, 8:16 pm

(May 17, 2019 07:43 PM)Kay Wrote: [\[CITE\]](#) There are certainly plenty of implementations out there for that you are trying to accomplish. I would suggest starting by googling some of those key terms, as well as looking on stack overflow and git repos. If you still need help after that, send me a PM.
i can't send you a message :/

