

CS2233 Assignment

CS18BTECH11018

1.

INPUT:

I am just taking the STRING and value of T only. I am not taking other functional inputs like Store,Sort,Print.

Reasoning:

I have created a linked list and took input of string, then I have appended all entries of string one by one into linked list (all in upper case as the input is case insensitive as per instruction given).

For part a, I traversed over the linked list and printed the root and last element.

Then for part b, I have traversed a whole linked list for count of particular alphabet and marked it as done once all entries of it are counted and we do this for all elements.

For part c, I have used the counts obtained from part b and sorted them in decreasing order and printed it. For part d, I have again used part b, counts and considered only T consecutive elements if $T > \text{count}$, else just printed.

I have used maps to store the count.

2.

INPUT:

I have considered that keys are given with spaces and pressing return will stop taking input of keys. For functionality input, input is to be given as mentioned in assignment and hitting return twice will terminate program.

Reasoning:

Created a structure for each node and added each key into graph using insert function. Predecessor is nothing but right most in the left side of the element and

vice versa for successor. I have printed no predecessor/successor in case there isn't any for given key.

Maximum for a root is right most element and vice versa for minimum. For common ancestor, i have compared both the elements with root and based on the outcome(root is greater than both the elements or less than one of them or smaller than both), I have recursed to left or right or return the answer.

Everything is done using structures and no other additional libraries are used.

3.

INPUT:

I have taken number of vertices as the first input as mentioned and then I have taken the input of edges in pattern as given in 4th question.

Once the input on edges is done, user needs to select one function from (a,b,c,d,e) using the alphabet itself and should pass the required arguments with spaces(for example a 1 or b or c 2 4 or d 3 or e 5 3). Hit return twice to exit.

I have written basic functions like

addEdge,neighbours,vertices,edgeWeight,containsVertex which can be used to get the required information once the input is given. I have implemented the graph using adjacency list with the help of vectors.

4.

INPUT:

I have taken number of vertices as the first input as mentioned and then I have taken the input of edges in pattern as given.Once taking input of edges is done then,

For functionality input enter

Find

SP source

For connected components i have used dfs and for shortest path I have used Dijkstra's algorithm. For dijstras I have used the priority queue from STL and used vectors for maintaining the graphs,performing search.

OUTPUT:

In the output

If there is no path from 1 vertex to another in a connected component because the graph is directed, I have printed that there is no path, for example for dijkstra on 1, i printed

No path from 1 to 0 in given directed graph! - As there was no path from 1 to 0 in that connected component.

Also find is to be entered only once and Sp can be used any number of times and enter 'x' or 'X' to exit the program.

5.INPUT:

I have taken number of vertices as the first input as mentioned and then I have taken the input of edges in pattern as given in 4th question.

Once input is done, hit return twice to see the output.

I have used the number of connected components to check whether we have an MST or not. If there are more than one connected components then i have printed no MST message , else I have printed the MST as per given output fashion in assignment.