**Dijkstra Sequence**

**Problem**

**Author:陈硕**

**Number:3220106406**

**Date:2023-12-4**

**I.Introduction**

Dijkstra’s algorithm is one of the most famous greedy algorithms to solve the **single source shortest path problems**.

In this algorithm, a set contains vertices included in shortest path tree is maintained. During each step, we find one vertex which is not yet included and has a minimum distance from the source, and collect it into the set. Hence step by step an ordered sequence of vertices, and let's call it **Dijkstra sequence.**

On the other hand, for a given graph, there could be more than one Dijkstra sequence.

**This project** **aims to check whether a given sequence is Dijkstra sequence or not.**

**II.Algorithm Specification**

Firstly,a two-dimension symmetric array **g[][]** is constructed, used to **store the whole graph**.In this array,**g[a][b]=m(m!=0) means there is an edge between vertice a and vertice b with the value of the edge being m**; g[a][b]=0 means that there is no edge between vertice a and vertice

**文本

描述已自动生成**

**Then,I design some functions to fulfill my object**.

**Function 1: dijkstra**

**This function is the core judge function**,if its return value is 1,the sequence is a Dijkstra sequence, otherwise not.In this function,I build

an array called **Set,storing the vertices we have included in** and another array called **Remainset**(built by Function 3),s**toring the vertices outside Set**. Everytime we meet a vertice in the given sequence,we first compute **min1**(computed by Function 2),**the minimum distance From Set to the vertice**.Then we compute **min2**(computed by Function 4),**the minimum distance from Set to Remainset**.If min1!=min2,it means the vertice don’t satisfy Dijkstra algorithm,then the sequence isn’t a Dijkstra sequence.If all vertices satisfy the requirements,the whole sequence is a Dijkstra sequence.

**文本

描述已自动生成**

**Function 2: MinFromSettoVertex**

**电脑屏幕的截图

描述已自动生成**This function aims to compute min1**.We can simply ergodic all the elements in Set to find the minimum distance**.Note that 0 can be considered as the minimum because 0 means that there is no edge between the two vertices!

**Function 3: Buildset**

This function aims to build Remainset. I set a mark”flag”.Then I ergodic Set and **mark all the elements that have appeared in Set with flag=0,meaning that they can’t appear in Remainset**.Hence I get the final Remainset.

**文本

描述已自动生成**

**Function 4: MinFromSettoRemainset**

**电脑截图

描述已自动生成**This function aims to compute min2.With Set and Remainset already built, **we can just simply ergodic all the vertices in Set and Remainset to find the smallest edge to connect the two set**. Also note that 0 can be considered as the minimum because 0 means that there is no edge between the two vertices!

**III.Testing Results**

|  |  |
| --- | --- |
| Testing data and results | Test for what |
|  | A graph with only a vertice |
|  | not be Dijkstra sequence |
|  | Be Dijkstra sequence. |
|  | Several sequences input at the same time |

**IV.Analysis and Comments**

**Let v be the the number of vertices in the graph.**

|  |  |  |
| --- | --- | --- |
| **function** | **Time complexity** | **Space complexity** |
| MinFromSettoVertice | O(v)  The number of elements in set varies from 1 to v,**hence it’s O(v)** | We build a two-dimension array g[][],storing the whole graph.**It takes v^2 space.**  We build two one-dimension arrays Set and Remainset,**both taking v spaces**  During the process of computing minimum distance,we first set a minimum,and compare the values computed a later with minimum,so **it takes O(1) space.**  Therefore,**the space complexity of the whole program is O(v^2).** |
| Buildset | O(v)  We actually ergodic all elements in Set to do the mark.**So it’s O(v)** |
| MinFromSettpRemainset | O(v^2)  When elements in Set and Remainset are both v/2,we do v^2/4 computations,which is the most.**So it’s O(v^2).** |
| dijkstra  (also the time complexity of the whole program) | O(v^3)  We ergodic all vertices in the sequence.In the loop,we call the three functions above,**so the whole time complexity is O(v)\*O(v^2)=O(v^3)** |

**In a nutshell,the time and space complexity of the whole program is respectively O(V^3) and O(v^2).**

From the analysis of time and space complexity, I can feel that there’s still much space for my project and algorithm to improve(especially the time complexity is too large).And I may use more efficient algorithms. Overall, I successfully finished my project,despite many difficulties.Hope that I can do better next time!

**V.Appendix**

**文本

描述已自动生成**

**文本

描述已自动生成**

**电脑屏幕的截图

描述已自动生成**

**文本

描述已自动生成**

**电脑截图

描述已自动生成**

**VI. Declaration**

I hereby declare that all the work done in this project titled”陈硕” is of my independent effort.