**Department of Computer Science & Engineering**

**Course: Design and Analysis of Algorithms**

**First Shift – Challenging Assignment Question**

**Max. Marks: 04**

**Rule: First Correct Solution: on or before 9th Feb 2016. [Individual Student]**

**Question:**

Given G=(V,E) be a directed graph. Let V1, V2, V3….Vn be the nodes in the graph. Graph “G” is ordered graph with following properties:

1. Each edge goes from a node with lower index to a node with higher index. Edge (Vi, Vj) with i<j.
2. Each node expect Vn has atleast one outgoing edge.

The length of a path is number of edges in it.

**Objective - 1**

The goal is to find the longest path that begins at V1 and ends at Vn.

**Objective – 2**

Code the following algorithm in any programming language and show that algorithm does not works correctly for the given problem.

Algorithm longest path()

{

W = V1

Length = 0

While (There is an edge out of the node W)

{

{For the value of “j” which is smallest as possible}

Select the edge (W, Vj)

Set W = Vj

Increase Length by 1

} end of while

Return L as length of longest path

}

**Objective – 3**

Give an efficient algorithm and code to achieve the goal of longest path.

M.B.Chandak

HoD CSE

5-2-2016

**Department of Computer Science & Engineering**

**Course: Design and Analysis of Algorithms**

**First Shift – Challenging Assignment Question**

**Max. Marks: 04**

**Rule: First Correct Solution: on or before 14th Feb 2016. [Individual Student]**

**Question:**

“Pretty Printing”

Input: Ragged text, for example

*Hello everyone.*

*This is assignment on Design and Analysis of Algorithm.*

*The last date of submission for assignment*

*is defined and fixed.*

*The assignment will be solved by individual student of VI semester CSE First shift.*

*There will be 16 such problems and first five submissions will be counted. You*

*can read the question, code, design test cases and submit*

Output expected

Hello everyone. This is assignment on Design and Analysis of Algorithm.

The last date of submission for assignment

is defined and fixed. The assignment will be solved by individual student of VI semester CSE First shift. There will be 16 such problems and first five submissions will be counted. You can read the question, code, design test cases and submit

**Observation:** The right margin is as “even” as possible.

**Idea:** Suppose the text consist of sequence of words W = {w1, w2, w3, wn} and each word is consist of Ci characters. The maximum length of each line is fixed by defined “right margin”. Each character size is also assumed as fixed.

The formatting function will partition W into lines. The words should be separated by “space” with except last word, so if Wj, Wj+1, Wk are words in one line then

The difference between left hand side and right hand side of above equation, will be called as slack of line, i.e., number of space unused with respect to right margin.

Give an efficient algorithm to find partition of set of word “W” into valid lines, so that sum of “squares” of slack of all lines including last line is minimized.

**Submit with Three Test Cases**

**Department of Computer Science and Engineering**

**Tutorial: SET - 1.1**

**Course: Design and Analysis of Algorithm**

**Course Coordinator: Dr.M.B.Chandak**

**Date: 9-02-2016**

**Topics: Multistage Graph, Travelling Salesman Problem, Longest Common Subsequence**

**Example: 1: Multistage Graph [Correct Solution 1 mark]**

**Convert the following matrix into Graph. Solve for Shortest path from Source node to Destination node**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| **1** | **0** | **12** | **9** | **6** | **11** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **2** | **0** | **0** | **0** | **0** | **0** | **9** | **0** | **11** | **0** | **0** | **0** | **0** |
| **3** | **0** | **0** | **0** | **0** | **0** | **8** | **4** | **9** | **0** | **0** | **0** | **0** |
| **4** | **0** | **0** | **0** | **0** | **0** | **2** | **5** | **7** | **0** | **0** | **0** | **0** |
| **5** | **0** | **0** | **0** | **0** | **0** | **9** | **5** | **11** | **0** | **0** | **0** | **0** |
| **6** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **8** | **0** | **2** | **0** |
| **7** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **11** | **4** | **5** | **0** |
| **8** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **7** | **2** | **11** | **0** |
| **9** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **6** |
| **10** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **4** |
| **11** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **8** |
| **12** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

**Example: 2 [Correct Solution: 1 mark]**

**Given an array of “n” real numbers, find the component of array such that sum of all element of component generates maximum value.**

**The complexity of presented algorithm should be O(n). [Cannot use nested loops]**

**Note: Array contains both positive and negative numbers.**

**Example: [31, -41, 59, 26, -53, 58, 97, -93, -23, 84]**

**Consider component = [59+26+(-53)+58+97] = 187**

**Department of Computer Science and Engineering**

**Tutorial: SET – 2.1**

**Course: Design and Analysis of Algorithm**

**Course Coordinator: Dr.M.B.Chandak**

**Date: 9-02-2016**

**Topics: Multistage Graph, Travelling Salesman Problem, Longest Common Subsequence**

**Example: 1: Travelling Salesman Problem [Correct Solution 1 mark]**

**Find the solution for travelling salesman problem, using given matrix.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **0** | **12** | **8** | **9** | **15** |
| **11** | **0** | **14** | **2** | **7** |
| **9** | **12** | **0** | **6** | **10** |
| **8** | **9** | **8** | **0** | **11** |
| **12** | **14** | **11** | **15** | **0** |

**Comment on the space complexity required for storing all the intermediate result during computing the path. [General equation for graph of “n” vertices]**

**Example 2: [Correct Solution 1 mark]**

**Convert the following matrix into Graph**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **1** | **0** | **10** |  |  |  |  | **8** |
| **2** |  | **0** | **12** |  |  |  |  |
| **3** |  |  | **0** | **9** |  |  |  |
| **4** |  |  |  | **0** | **6** |  |  |
| **5** |  |  |  |  | **0** | **14** |  |
| **6** |  |  |  |  |  | **0** | **4** |
| **7** |  |  |  |  |  |  | **0** |

**Comment on Degree of each vertex.**

**Add edges in the graph, such that each component of graph becomes triangle. The total cost of all newly added edges should be minimum. There is no restriction on number of edges to be added in the graph.**

**Department of Computer Science and Engineering**

**Tutorial: SET – 3.1**

**Course: Design and Analysis of Algorithm**

**Course Coordinator: Dr.M.B.Chandak**

**Date: 9-02-2016**

**Topics: Multistage Graph, Travelling Salesman Problem, Longest Common Subsequence**

**Example: 1: Longest Common Subsequence [Correct Solution 1 mark]**

**For the following two strings, find out the longest common subsequence. Generate matrix. Write logic to find out number of subsequences in the LCS.**

**String 1: S E Q U E N T I A L**

**String 2: E X P O N E N T I A L**

**Example 2: [Correct Solution 1 mark]**

**Given an array of integers, write algorithm to find out longest increasing subsequence. For example:**

**45, 23, 9, 3, 99, 108, 76, 12, 77, 16, 18, 4**

**The longest increasing sequence is: 3, 12, 16, 18 and length is 4.**

**Department of Computer Science and Engineering**

**Tutorial: SET - 1.2**

**Course: Design and Analysis of Algorithm**

**Course Coordinator: Dr.M.B.Chandak**

**Date: 9-02-2016**

**Topics: Multistage Graph, Travelling Salesman Problem, Longest Common Subsequence**

**Example: 1: Multistage Graph [Correct Solution 1 mark]**

**Convert the following matrix into Graph. Solve for Shortest path from Source node to Destination node**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| **1** | **0** | **7** | **12** | **15** | **11** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **2** | **0** | **0** | **0** | **0** | **0** | **14** | **10** | **0** | **0** | **0** | **0** | **0** |
| **3** | **0** | **0** | **0** | **0** | **0** | **8** | **4** | **9** | **0** | **0** | **0** | **0** |
| **4** | **0** | **0** | **0** | **0** | **0** | **7** | **4** | **6** | **0** | **0** | **0** | **0** |
| **5** | **0** | **0** | **0** | **0** | **0** | **11** | **12** | **8** | **0** | **0** | **0** | **0** |
| **6** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **8** | **0** | **2** | **0** |
| **7** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **10** | **4** | **5** | **0** |
| **8** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **9** | **12** | **5** | **0** |
| **9** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **6** |
| **10** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **5** |
| **11** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **9** |
| **12** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

**Example: 2 [Correct Solution: 1 mark]**

**Given an array of “n” real numbers, find the component of array such that sum of all element of component generates maximum value.**

**The complexity of presented algorithm should be O(n). [Cannot use nested loops]**

**Note: Array contains both positive and negative numbers.**

**Example: [31, -41, 59, 26, -53, 58, 97, -93, -23, 84]**

**Consider component = [59+26+(-53)+58+97] = 187**

**Department of Computer Science and Engineering**

**Tutorial: SET – 2.2**

**Course: Design and Analysis of Algorithm**

**Course Coordinator: Dr.M.B.Chandak**

**Date: 9-02-2016**

**Topics: Multistage Graph, Travelling Salesman Problem, Longest Common Subsequence**

**Example: 1: Travelling Salesman Problem [Correct Solution 1 mark]**

**Find the solution for travelling salesman problem, using given matrix.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **0** | **6** | **11** | **12** | **10** |
| **11** | **0** | **12** | **2** | **9** |
| **9** | **14** | **0** | **6** | **8** |
| **8** | **9** | **8** | **0** | **11** |
| **12** | **13** | **7** | **14** | **0** |

**Comment on the space complexity required for storing all the intermediate result during computing the path. [General equation for graph of “n” vertices]**

**Example 2: [Correct Solution 1 mark]**

**Convert the following matrix into Graph**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **1** | **0** | **10** |  |  |  |  | **8** |
| **2** |  | **0** | **12** |  |  |  |  |
| **3** |  |  | **0** | **9** |  |  |  |
| **4** |  |  |  | **0** | **6** |  |  |
| **5** |  |  |  |  | **0** | **14** |  |
| **6** |  |  |  |  |  | **0** | **4** |
| **7** |  |  |  |  |  |  | **0** |

**Comment on Degree of each vertex.**

**Add edges in the graph, such that each component of graph becomes triangle. The total cost of all newly added edges should be minimum. There is no restriction on number of edges to be added in the graph.**

**Department of Computer Science and Engineering**

**Tutorial: SET – 3.2**

**Course: Design and Analysis of Algorithm**

**Course Coordinator: Dr.M.B.Chandak**

**Date: 9-02-2016**

**Topics: Multistage Graph, Travelling Salesman Problem, Longest Common Subsequence**

**Example: 1: Longest Common Subsequence [Correct Solution 1 mark]**

**For the following two strings, find out the longest common subsequence. Generate matrix. Write logic to find out number of subsequences in the LCS.**

**String 1: S U B S E Q U E N C E**

**String 2: U S A M E N I C E**

**Example 2: [Correct Solution 1 mark]**

**Given an array of integers, write algorithm to find out longest increasing subsequence. For example:**

**45, 23, 9, 3, 99, 108, 76, 12, 77, 16, 18, 4**

**The longest increasing sequence is: 3, 12, 16, 18 and length is 4.**