|  |  |
| --- | --- |
| COMP 2121 **DISCRETE MATHEMATICS** | Assignment 3  Fall 2024 |

|  |  |  |
| --- | --- | --- |
| Name:???  A0??? | Set: 2? | |
| Name:??? A0??? | Set: 2? | |
|  |  | |
| Section | Total | Actual |
| Question 1 | 12 |  |
| Question 2 | 15 |  |
| Question 3 | 6 |  |
| Question 4 | 6 |  |
| Question 5 | 6 |  |
| Total | 45 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| C:\Users\A00141222\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\ARBZC6KC\MC900432530[1].png | | | Instructions |
|  |  |  | * Assignment must be done using Microsoft Word - type your work in this document. Handwritten assignments will not be marked. * Figures can be done by hand. |
|  |  |  |
|  |  |  |
|  |  |  | * All work/steps need to be shown in **Q1, Q2.** * Figures need to be provided in **Q2** * No explanations are required in **Q3, Q4, Q5** |
|  |  |  |
|  |  |  |
|  |  |  | * The assignment must be done in pairs. |
|  |  |  |
|  |  |  |
|  |  |  | * **PRINT** the completed assignment – you are handing in a paper copy. |
|  |  |  |
|  |  |  |
|  |  |  | * **Due** at the beginning of the **Lecture** on **Nov 27, 2024**. * No late assignments will be accepted. * Electronic copies will not be accepted. |
|  |  |  |
|  |  |  |

**Q1)** Use mathematical induction to show that MysteryFunction(n) returns , for .

MysteryFunction (positive integer n)

i = 1

j = 13

**while** i ≠ n + 2 **do**

i = i + 1

j = 4j - 21i - 12

**end while**

return j

All work/steps need to be shown.

**Q2**) Consider the graph below. Find the number of spanning trees having exactly one vertex with a degree of 5.



Instructions:

* Only answers that provide **several** figures helping, explaining, and supporting the solution will be considered.
* Every figure requires clear explanation(s).

The figures can be done by hand but neatly—insert a picture into the document. Alternatively, reuse the figure above in Visio and add details.

When writing explanations, it may be helpful to imagine the audience as a student in 2121 who is struggling a little bit with the material.

**Explanations:**

1. **provide the final formula**, put everything together.

**2) evaluate the formula**.

**Q3)** Run modified DFS algorithm starting at **vertex 3**.



dfs(vertex v)

{

for each neighbor w of v //*in the increasing order*

if w is unvisited

{

visit(w);

print v

dfs(w);

print w

}

}

Show the following and delete empty cells at the end:

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| 3 |
|  |

1) Return Stack

2) Printed on the Screen:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Q4)** Run the BFS algorithm starting at **vertex 11** and show its spanning tree. Give a label to each edge according to the order in which it is added to the spanning tree (i.e. edge that is added first label 1, edge that is added second label 2, etc.). Show queue Q and delete empty cells at the end.

**Instructions:** You must use the algorithm presented in the class. Yes, you can label and highlight the edges by hand. Alternately, import the figure in Visio, label it, highlight the edges, and then paste it back into the document.



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 11 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Queue Q:

**Q5**) We define “SomeOrder” tree Traversal Algorithm as follows:

SomeOrder Traversal Algorithm:

Step 1: (start)

Go to the root.

Step 2: (go right)

Go to the **right** subtree, if one exists, and do a SomeOrder traversal.

Step 3: (go right)

Go to the **right** subtree, if one exists, and do a SomeOrder traversal.

Step 4: (go left)

Go to the **left** subtree, if one exists, and do a SomeOrder traversal.

Step 5: (visit)

Visit the root.



For the tree shown, list the vertices according to the SomeOrder traversal.