



**Eduvos**  
Your Education. Your Future.

Eduvos (Pty) Ltd (formerly Pearson Institute of Higher Education) is registered with the Department of Higher Education and Training as a private higher education institution under the Higher Education Act, 101, of 1997. Registration Certificate number: 2001/HE07/008.

Date: Wednesday, 30 October 2024, 2:28 PM

# Data Structures and Algorithms in C# Assessments

## Project

### 1. Project

Faculty Name:	Information Technology
Module Code:	ITDCA2-44
Module Name:	Data Structures and Algorithms in C#
Copy Editor:	Kyle Keens
Total Marks:	100
Submission Date:	Week 6

---

This module is presented on NQF level 6.

5% will be deducted from the student's project mark for each calendar day the project is submitted late, up to a maximum of three calendar days. The penalty will be based on the official campus submission date.

Projects submitted later than three calendar days after the deadline or not submitted will get 0%. <sup>[1]</sup>

This is an individual project.

---

**This project contributes 40 % towards the final mark.**

[1] Under no circumstances will projects be accepted for marking after the projects of other students have been marked and returned to the students.

## 2. AI Checklist and Declaration

Before you submit an assignment, you should be able to confidently and honestly make all the below statements. For group work, you can also review the list, together, to hold one another accountable.

- I confirm that my submission reflects my personal learning, knowledge, skills, and understanding.
- If AI tools were employed for generating any part of this assignment (even in the drafting/research phase), I have referenced the use of AI in the text and/or declared the use of AI. I am willing to discuss the process and its contribution to my learning.
- I am aware that the lecturer may request a demonstration of my learning, such as explaining choices in approach, research, and the content I am submitting.
- I am aware that, if I did use AI in any phase of preparing this submitted work, it is recommended that I save a copy of the relevant chat history (prompts and answers), as this will help me demonstrate my writing/work process to my lecturer, if I am asked to do so.
- I have read the assignment instructions on whether AI tools are prohibited for this assignment, and if they are prohibited, I can confirm that I did not use AI tools.
- I understand that failure to agree to these terms may be deemed unethical, potentially leading to disciplinary action. I understand my responsibility for the integrity of my work, including seeking clarification from academic staff and adhering to instructions.

It is essential to acknowledge your use of ChatGPT or other generative AI in your learning. If you use ChatGPT or other generative AI to help you generate ideas or plan your process, you should still acknowledge how you used the tool, even if you don't include any AI-generated content in the assignment.

**Please note:** The following guiding questions that you will be asked in an AI declaration questionnaire below this assignment brief.

### AI Declaration

**It is compulsory to complete this AI declaration for each of your assignment submissions.**

I carefully read the assignment instructions, and the extent to which AI may be used for the assignment.

I used the following AI system(s)/tool(s):

I used it for the following:

If I quoted or paraphrased an AI output, I have referenced the relevant tool, version, and the date I used the tool.

I still consider this work my own. (i.e., I have not outsourced the final product, or significant portions of it, to AI tools/systems).

If required, I can defend my argument/perspective, explain my choices and approach, and can show that I am knowledgeable about the details of my work.

For further guidance on the use of AI at Eduvos, please refer to the AI FAQ glossary. You will locate the FAQs in the Artificial Intelligence tile on the myDocuments page of myLMS.

### 3. Instructions to Students

1. Please ensure that your answer file (where applicable) is named as follows before submission: **Module Code – Assessment Type – Campus Name – Student Number**.
2. Remember to keep a copy of all submitted assignments.
3. All work must be typed.
4. Please note that you will be evaluated on your writing skills in all your assignments.
5. All work must be submitted through Turnitin. The full originality report will be automatically generated and available for the lecturer to assess. Negative marking will be applied if you are found guilty of plagiarism, poor writing skills, or if you have applied incorrect or insufficient referencing. (See the "instructions to students" book activity before this activity where the application of negative marking is explained.)
6. You are not allowed to offer your work for sale or to purchase the work of other students. This includes the use of professional assignment writers and websites, such as Essay Box. You are also not allowed to make use of artificial intelligence tools, such as ChatGPT, to create content and submit it as your own work. If this should happen, Eduvos reserves the right not to accept future submissions from you.

## 4. Section A

---

### Section A

---

#### Learning Objective

1. Acquire the skill to interpret data structures and algorithms.
2. Implement data structures in C#.
3. Develop the skill to select a suitable data structure/algorithm and effectively justify your decision.

#### Project Topic

Graphs

#### Scope

Weeks 1 to 6

#### Technical Aspects

Programming in C#

## 4.1. Question 1

---

### Question 1

**25 Marks**

---

Write a graph implementation in C#. Remember that graphs also require a Node and Edge class, which you must implement as well. In your implementation, graphs are not directional, but they are weighted. Nodes are labelled with integers.

---

End of Question 1

## 4.2. Question 2

---

### Question 2

**25 Marks**

---

Expand on Question 1 by implementing a depth-first traversal and breadth-first traversal of the graph.

You will also implement a function to read in graphs from text files. The expected format of text files are:

```
<number of nodes>  
  
<number of edges>  
  
<node from> <node to> <weight> // Edge 1  
  
<node from> <node to> <weight> // Edge 2  
  
<node from> <node to> <weight> // Edge 3  
  
...
```

You may include this in the Graph class from Question 1. Ensure you clearly comment which code is applicable to Question 2. You will be penalised if you don't.

---

End of Question 2

## 5. Section B:

---

### Section B

---

**Learning Objective**

1. Acquire the skill to interpret data structures and algorithms.
2. Implement data structures in C#.
3. Develop the skill to select a suitable data structure/algorithm and effectively justify your decision.

**Project Topic**

Minimum spanning trees

**Scope**

Weeks 1 to 6

**Technical Aspects**

1. Programming in C#

**Study the scenario and complete Questions 3 and 4 that follow:**

## Network Optimisation

A new telecommunications company in South Africa wants to link cities with fibre optic cables. As this is a new company, they want to keep costs low, and plan to expand their network later. They have a list of cities they want to connect to their network. They have also estimated the cost of laying cables between different cities, but, due to permission requirements, a direct link is only possible between some cities. However, thanks to network routers, a direct link between every city is not required – the router can calculate an indirect route between cities.

The company needs to calculate a configuration that will connect all cities in the network at the lowest cost. An internal developer mentioned 2 such algorithms that can calculate this: Kruskal's algorithm and Prim's algorithm. To help them decide which algorithm to use for this and future endeavours, the company wants to know which will run faster for their use-case.

### 5.1. Question 3

#### Question 3

**20 Marks**

Write a program that will calculate the minimum spanning tree of a graph using Kruskal's algorithm. We also need to know how fast this algorithm will run, so you may use `DateTime` to calculate this. Note we require the runtime of the actual Kruskal's algorithm only, not the time taken to add nodes and edges to the graph.

You may add this code to the graph class you have written earlier. Comment clearly which code you have written for this question. You will be penalised if you don't.

Once you have written your program, run it against the provided sample graph (`graph.txt`) 10 times, to obtain 10 runtimes. Keep record of these runtimes for the next question.



End of Question 3

## 5.2. Question 4

---

### Question 4

**30 Marks**

---

Write a program that will calculate the minimum spanning tree of a graph using Prim's algorithm. We also need to know how fast this algorithm will run, so you may use DateTime to calculate this. Note we require the runtime of the actual Prim's algorithm only, not the time taken to add nodes and edges to the graph.

You may add this code to the graph class you have written earlier. Comment clearly which code you have written for this question. You will be penalised if you don't.

Once you have written your program, run it against the provided sample graph (graph.txt) 10 times, to obtain 10 runtimes.

Plot the runtimes you have calculated in Question 3 (the Kruskal runtimes) and the runtimes calculated here (the Prim runtimes) onto 2 box and whisker diagrams.

Write a brief paragraph to explain your findings, justifying any observations made.

---

End of Question 4

