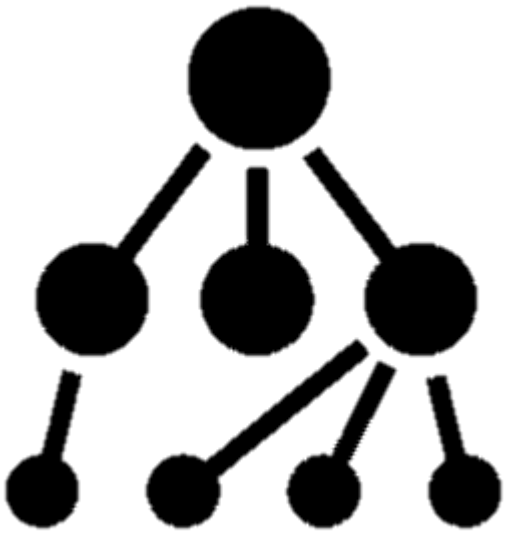
Data Structures & Algorithms – Final Report

A technical report covering the applications of hash tables & graphs.



By: Mark Lambert (C00192497), Darius Herdes (C00296548)

Course: Software Development (CW_KCSOF_B)

Submission Date: 10/12/2024

Ollscoil Teicneolaíochta an Oirdheiscirt
South East Technological University

Table of Contents Page

Introduction

This report wash commissioned by Áine Byrne as a final project for the Data Structures & Algorithms module. The purpose of the report is to display an understanding of the implementation of a hash application, a graph application including a coded protype to show said implementation. The report should also show one's ability to perform as part of a group dynamic.

Hash Table Application

What is a Hash Table?

A hash table is a data structure wherein a piece of information is stored in an encrypted manner after performing what's known as a "hash function" on some input. The hash function produces an index position for one single unit of information to be stored in the hash table, which is implemented as an array.

Application

The hash table application will take in a username (max characters: 6) and create a unique index for each username. The key is calculated by accumulating the total decimal ASCII value of the characters and performing a modulo of 20. In the case of a collision, a linear probe will be considered.

For this implementation, the final hash table will be of length 20 and the hash function will be % 20.

Example hash function equation

$$mlamb - 521 \,\%\, 20 = 1$$

Table of Data

The following table of data shows an example of 8 username inputs, some with uppercase characters.

Username	ASCII	Index (after hash function %20)
mlamb	521	1
darius	648	8
AdAm	339	19
hannah	662	2
Aaron	497	17
frank	529	9
james	528	8
daliah	611	11

Diagram of Hash Table Produced

Below is a diagram visualising what the hash table will look like with each username in their respective index as well as any collisions which occurred.

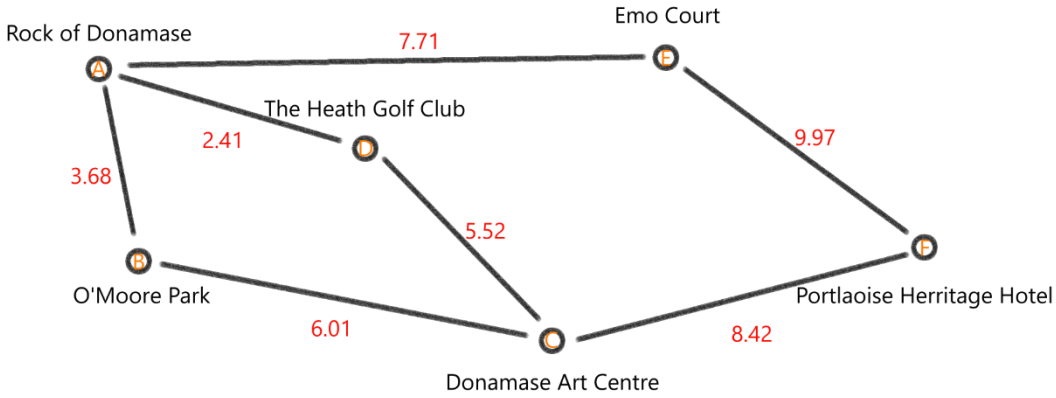
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Username		mlamb	hannah						darius	frank	james	daliah						Aaron		AdAm
Collision											2									

Graph Application

What is a Graph?

An **adjacency matrix** is a square matrix used to represent a graph, where rows and columns correspond to nodes. Each entry indicates whether an edge exists between a pair of nodes: 1 (or a weight) for an edge and 0 for no edge. It is widely used for analyzing graph structures in both directed and undirected graphs.

The image shows a graph connecting six points of interest in Portlaoise, labeled A to F, where nodes represent locations (e.g., A = Rock of Dunamase) and edges are labeled with distances in kilometers. Below the graph is an adjacency matrix, where rows and columns correspond to locations, and each cell shows the distance between two points. A value of 0 indicates no direct connection. The graph and matrix provide a clear visualization of the distances and connections between the locations.



	A	B	C	D	E	F
A	0	3.68	0	2.41	7.71	0
B	3.68	0	6.01	0	0	0
C	0	6.01	0	5.52	0	8.42
D	2.41	0	5.52	0	0	0
E	7.71	0	0	0	0	9.97
F	0	0	8.42	0	9.97	0

Note: An adjacency matrix should be symmetrical via the diagonal if a graph is undirected.

Graph Implementation

What data structure? A 2D array of equal length rows as it has columns will be the chosen data structure to implement this application.

It should be noted that using a 2D array implies the use of an **adjacency matrix** versus the use of an **edge list.**

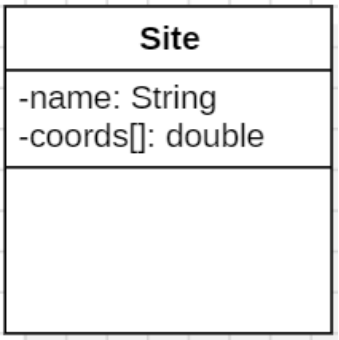
- Initially the adjacency matrix should **not** be populated. Meaning, no nodes have an associated edge. (Represented as a **0**)
- The graph is **weighted** and **undirected**. Edges are represented in weights of kilometres greater than 0.

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

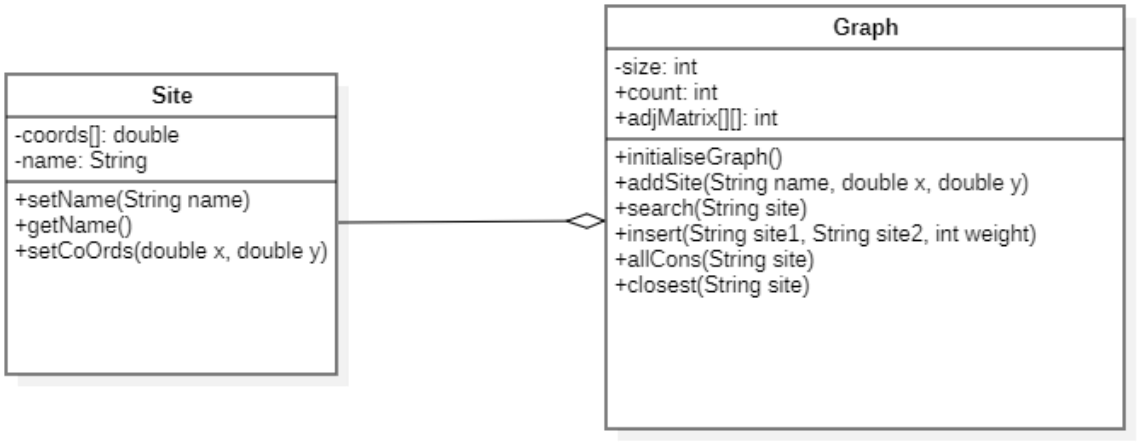
Adjacency matrix representing a graph with no edges between nodes

Defining Tourist Sites -"Each [tourist site] stored will contain the name of the site and the co-ordinates…" – as set out in the rubric.

- To store the site information each site will be represented as its own **Object.** This allows easy access to store and read site names and co-ordinates.
- To **add** a site to the graph, a name of type String and x and y coordinates of type double should be supplied.



The Graph object will communicate with the Site object via a form of basic aggregation. Each Graph object will hold the sites via an array of type Site.

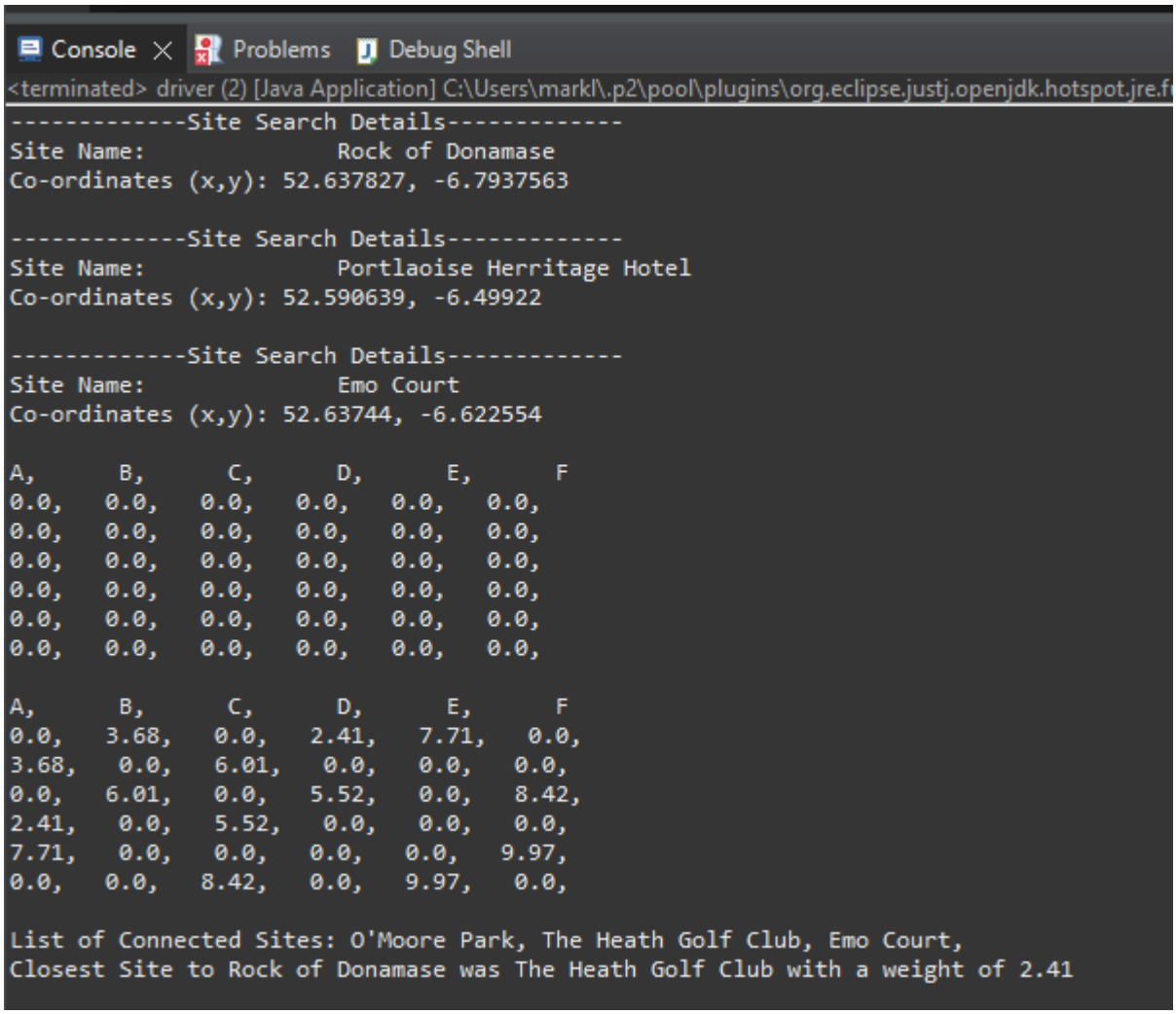


Pseudocode for Algorithms

```
//Explanation of un-declared helper variables
count = number of current sites (nodes) in the graph
sites[] = array of site objects (nodes) in the graph


int isSite(String siteName)
{
    for(i = 0; i < count; i++)
    {
        if(sites[i].getName().equals(siteName))
        {
            return sites[i].getIndex()
        }
    }

    return -1

}
```

```
Algorithm allCons(String site)
{
    index = isSite(site)

    if(isValid)
    {
        for(col = 0; col < count; col++)
        {
            if(graph[index][col] > 0)
                {
                    output sites[col]
                }
            }
        }
        else
            output "Invalid site index"
}


Algorithm closest(Site site)
{
    index = isSite(site)

    if(isValid)
    {
        for(col = 0; col < count; col++)
        {
            if(graph[index][col] < smallest AND adjMatrix[index]col != 0)
            {
                smallest = graph[index][col]
            }
        }
    }

    output smallest
}
```

Sample Output



Minimum Spanning Tree - Dijkstra's Algorithm

Step	Choice	Rock of Donamase	O'Moore Park	Donamase Art Centre	The Heath Golf Club	Emo Court	Portlaoise Heritage Hotel
Init.		0	∞	∞	∞	∞	∞
1.	Rock of Donamase	-	3.68	-	2.41	7.71	-
2.	The Heath Golf Club	-	3.68	7.93	2.41	7.71	-
3.	O'Moore Park	-	3.68	7.93	-	7.71	-
4.	Emo Court	-	-	7.93	-	7.71	17.68
5.	Donamase Art Centre	-	-	7.93	-	-	14.43
6.	Portlaoise Heritage Hotel	-	-	-	-	-	14.43

Acknowledge, Describe, Evidence Document

Include a completed version of this document as an appendix to any submitted work.

Acknowledge

☒ I did NOT use any AI Technology or online resource.

I acknowledge the use of <insert AI system(s) and link> or <weblink> for the following purposes:

☐ to generate materials for background research and self-study in the drafting of this assessment.

☐ to generate materials that were included within my final assessment in modified form.

Describe

Please provide a short summary of how you used generative AI/website in your assignment. You may wish to include the following information:

What prompts did you use?

What outputs did you generate?

How did you use/adapt/develop the outputs?

Summary:

Evidence

Please provide evidence of the outputs that you generated by copying and pasting below or by providing a screenshot.

Generative AI system/Website:

Prompt:

Output:

Declaration

☒ I confirm that no content created by generative AI technologies or website research has been presented as my own work.