

DFS Assignment Report

1808257

by Isobel Sutcliffe

Submission date: 07-Jan-2022 09:05PM (UTC+0000)

Submission ID: 167918472

File name: 12097_Isobel_Emma_Sutcliffe_DFS_Assignment_Report_1808257_393826_1929943471.docx
(827.9K)

Word count: 2637

Character count: 12534

AI and Machine Learning

ACCA6011 – G Dickers

Search

Isobel Sutcliffe – 1808257

1808257@student.uwstd.ac.uk

Content

Task 1	1
1.1 Depth First Search	1
1.2 User Guide	3
1.3 Program listing (Code and Mapping Text Files).....	10
1.4 Testing Schedule.....	11
Task 2	14
Task 3	18
References	21
Appendices	22
Appendix 1	22

Task 1

1.1 Depth First Search

A Depth First Search (DFS) algorithm is an algorithm that always expands to the deepest node when searching a tree. [1]

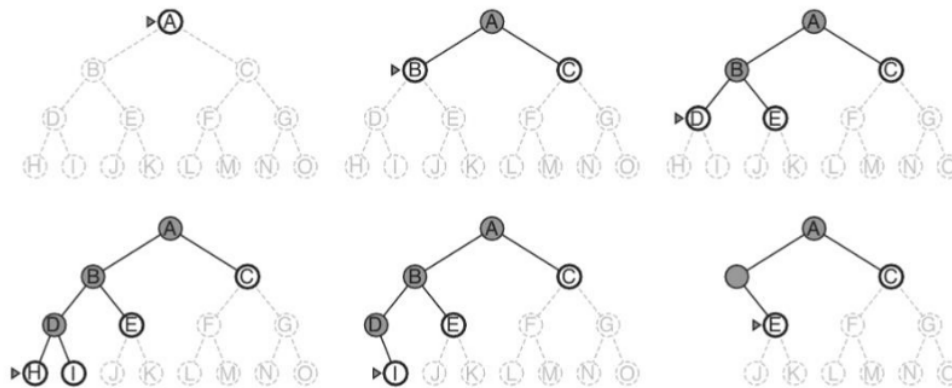


Image of a DFS [1]

From the image above it can be seen that from A the algorithm will note points B and C, going to B first. From B it will then note points D and E, going to D from B. Once at D it will note H and I, moving to H in the first instance and then I. Once it has explored the lowest levels it will then proceed back up to E and move down to J and then K.

The whole process will be:

A -> B -> D -> H -> I -> E -> J -> K -> C -> F -> L -> M -> G -> N -> O

DFS is based on Last In First Out (LIFO) rather than First In First Out (FIFO) which Breadth First Search (BFS) depends on. This means that the most recently generated node is chosen to be explored, usually one deeper than its parent, as so going deeper down until it can go no further. [1]

When creating the program to search for the pathway around the map stack was included using `#include <stack>`. A stack is a type of container that allows LIFO working, where a new element is added to the top of the stack and the next element is removed from the same end. [2]

This allows the programmer to perform the following tasks once the constructor for the stack is set up: [2]

- `empty()` – determines if the stack is empty;
- `size()` – determines the number of items within the stack, returning a value;
- `top()` – returns the top element within the stack;
- `push()` – adds an element at the top of the stack;
- `pop()` – deletes the uppermost element on the stack.

The data structure used is known as a *strut*. A strut is used to store different data elements together, it is a user defined data type. [3] In the program created it is used to store the x and y coordinates as integers for the position the agent takes as well as the character value, if available, in the space, before the agent visits the space.

1.2 User Guide

When running the program, the user will have the ability to choose whether they want to:

- Upload a file;
- Print the uploaded map;
- Manually walk through the map themselves;
- Allow the DFS algorithm to walk through the map for them.

```
##### MENU #####
# 1: Load a file #
# 2: Print map   #
# 3: Random walk #
# 4: Find Signs  #
# 5: End program #
##### #### #####
Please enter your choice (1 to 4):
```

Figure 1 - Image of Menu Options

Option 1 – loading a file

```
##### MENU #####
# 1: Load a file #
# 2: Print map   #
# 3: Random walk #
# 4: Find Signs  #
# 5: End program #
##### #### #####
Please enter your choice (1 to 4): 1
Please enter file name, with file extension for completeness
```

Figure 2 - Image showing request for filename to be uploaded

- User must enter 1 to upload a file
- User will then be asked to input the file name with the file extension type as this is the full name of the file to be uploaded;
- In this example we are using “floor0.txt”;
- Once file has been uploaded an image of the file and a message stating “file loaded” will appear as in Figure 3 below:

```

##### MENU #####
# 1: Load a file #
# 2: Print map   #
# 3: Random walk #
# 4: Find Signs  #
# 5: End program #
##### #####
Please enter your choice (1 to 4): 1
Please enter file name, with file extension for completeness
floor0.txt
*****
S          A *
*****
*C*          *
* ***** *
* *          *
* *****
*          B*      *D*
* ***** *
* *          * *
* ***** *
* *****
* *****
* *****
* *****
*E          F*
*****
file loaded

##### MENU #####
# 1: Load a file #
# 2: Print map   #
# 3: Random walk #
# 4: Find Signs  #
# 5: End program #
##### #####
Please enter your choice (1 to 4):

```

Figure 3 - Image Showing File Loaded

- User will be returned to the menu options.

Option 2 – print map

```

##### MENU #####
# 1: Load a file #
# 2: Print map    #
# 3: Random walk  #
# 4: Find Signs   #
# 5: End program  #
##### #####
Please enter your choice (1 to 4): 2
*****
S          A *
*****
*C*          *
* ***** *
* *          *
* ***** *
*   B*      *D*
* ***** *
* *          *
* ***** *
* ***** *
* ***** *
* ***** *
*E          F*
*****

##### MENU #####
# 1: Load a file #
# 2: Print map    #
# 3: Random walk  #
# 4: Find Signs   #
# 5: End program  #
##### #####
Please enter your choice (1 to 4):

```

Figure 4 - Image Showing the map printed

- User must enter 2 to produce a copy of the map;
- User will see an image of the map as requested and will be returned to the options menu.

Option 3 – random walk

```

##### MENU #####
# 1: Load a file #
# 2: Print map   #
# 3: Random walk #
# 4: Find Signs  #
# 5: End program #
##### #####
Please enter your choice (1 to 4): 3

```

Figure 5 - Image Showing User Selection of 3 for Random Walk

- User must enter 3 to allow a random user walk of the map;
- Map will appear as in Figure 6 below:

```

*****
S      A  *
*****
*C*      *
*  *      *
*  *      *
*  *      *
*  B*    *D*
*  *      *
*  *      *
*  *      *
*  *      *
*  *      *
*  *      *
*  *      *
*E      F*
*****
Please use z to exit at any time.
Please enter either a, d, w, s to move position

```

Figure 6 - Map Awaiting Instructions

- User can use either of the following keys to navigate the map:
 - d – allows user to go across to the right one;
 - a – allows user to go across to the left one;
 - w – allows user to go up one on screen;
 - s – allows user to go down one screen.
 - z – allows the user to exit from the random walk screen at any point and return to the Option menu.

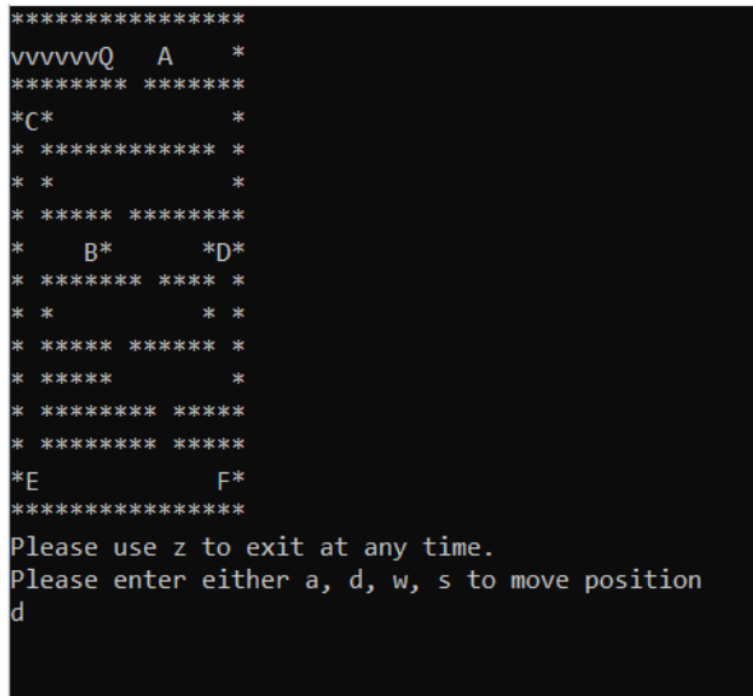


Figure 7 – Image of the AI Agent "Q", visiting the route - "v"

Option 4 – find signs – allow the AI to walk the paths

- User must enter 4 to allow the AI to walk the path on the map;
- User will be asked to enter the department or sign that they are looking for on the map:

```
##### MENU #####
# 1: Load a file #
# 2: Print map   #
# 3: Random walk #
# 4: Find Signs  #
# 5: End program #
#####
Please enter your choice (1 to 4): 4
Current position of the AI Agent is: (0 , 0 )
Start position for the AI Agent will be: (0 , 1)
Is there a particular department you would like to visit, please enter choice (if not please enter Z):
```

Figure 8 – Image of the Initial Input request

- User can either enter A, B, C, D, E, F for a department in this particular map;

```

##### MENU #####
# 1: Load a file #
# 2: Print map #
# 3: Random walk #
# 4: Find Signs #
# 5: End program #
##### #####
Please enter your choice (1 to 4): 4
Current position of the AI Agent is: (0 , 0 )
Start position for the AI Agent will be: (0 , 1)
Is there a particular department you would like to visit, please enter choice (if not please enter Z):
E

```

Figure 9 – Image of the User Inputting the Value E for Exploration

- AI Agent Q, fires through the pathway, using v to confirm where it has visited until it reaches the value E;
- Once E has been found it advises the user of the find and then proceeds to list the co-ordinates of E along with the coordinates explored in its pathway to locating it.

```

*****
vvvvvvvv A *
*****v*****
*C*vvvvvvvvvv*
* *****v*
* * vvvvvvvv*
* *****v*****
* B*vvv *D*
* *****v*****
* * vvv * *
* *****v*****
* *****vvv *
* *****v*****
* *****v*****
*Evvvvvvvv F*
*****
Department FOUND
Length of the path way stack is : 56
Pathway elements are : (1, 14, E)
Pathway elements are : (2, 14, )
Pathway elements are : (3, 14, )
Pathway elements are : (4, 14, )
Pathway elements are : (5, 14, )
Pathway elements are : (6, 14, )
Pathway elements are : (7, 14, )
Pathway elements are : (8, 14, )
Pathway elements are : (9, 14, )

```

Figure 10 – Image of AI Agent Finding E along with E's Coordinates and Explored Pathway

```
Pathway elements are : (7, 1, )
Pathway elements are : (6, 1, )
Pathway elements are : (5, 1, )
Pathway elements are : (4, 1, )
Pathway elements are : (3, 1, )
Pathway elements are : (2, 1, )
Pathway elements are : (1, 1, )
Pathway elements are : (0, 1, )
All viable positions explored
Length of the path way stack is : 0

##### MENU #####
# 1: Load a file  #
# 2: Print map    #
# 3: Random walk  #
# 4: Find Signs   #
# 5: End program  #
##### ####
Please enter your choice (1 to 4):
```

Figure 11 – Image of the End of E's Pathway Exploration

Any complaints about this user guide please do not hesitate to contact the team on
"Mapping@Mapping.com"

1.3 Program listing (Code and Mapping Text Files)



DFS Code.txt



Test1.txt



Test2.txt



floor0.txt

1.4 Testing Schedule

All tests are going to be performed on a 5 by 5 square to avoid delays:

Test	Test Description	Expected Outcome	Test Image	Passed/Failed
1	Select 1 – Load a file	User is asked to upload a file	 Test1.png	✓
2	Can user upload Test2.txt	***** S * *** * *A* * ***** File Uploaded	 Test2.png	✓
3	Can user upload Test1.txt after Test2.txt	***** * N * *W E* * S * ***** File uploaded	 Test3.png	✓
4	Is the newly uploaded map displayed after entering 2 – Print map (Test1.txt)	***** * N * *W E* * S * *****	 Test4.png	✓
5	Select 3 - Random Walk (Test1.txt)	***** * N * *W E* * S * ***** With output asking user to enter a direction value.	 Test5.png	✓
6	Enter d to move to the right on screen (Test1.txt)	The value Q used to identify the Agent moves to the right.	 Test6.png	✓
7	Enter a to move to the left on screen (Test1.txt)	The value Q used to identify the Agent moves to the left.	 Test7.png	✓

Test	Test Description	Expected Outcome	Test Image	Passed/Failed
8	Enter s to move down on screen (Test1.txt)	The value Q used to identify the Agent moves down.	 Test8.png	✓
9	Enter w to move up on screen (Test1.txt)	The value Q used to identify the Agent moves up.	 Test9.png	✓
10	Enter z to exit back to main menu (Test1.txt)	Option menu appears	 Test10.png	✓
11	Select 4 – Find Sign (Test1.txt)	Series of outputs advising of the current position along with the visitor position, asks the user if there is a particular department they would like to find.	 Test11.png	✓
12	Find S (down) (Test1.txt)	Identifies that it has found S and provides the coordinates to get there.	 Test12.png	✓
13	Find W (left) (Test1.txt)	Identifies that it has found W and provides the coordinates to get there.	 Test13.png	✓ Does not take a direct route
14	Find N (up) (Test1.txt)	Identifies that it has found N and provides the coordinates to get there.	 Test14.png	✓ Does not take a direct route
15	Find E (right) (Test1.txt)	Identifies that it has found E and provides the coordinates to get there.	 Test15.png	✓ Does not take a direct route

Test	Test Description	Expected Outcome	Test Image	Passed/Failed
16	No department (Test1.txt)	Walks through the whole map – no department to find	 Test16.png	 Doubling stacking positions for visiting – backtracking needs to be considered – child and parent relationships

Task 2

Clustering Algorithms

What is a Clustering Algorithm?

A clustering algorithm is an algorithm that is unsupervised task that looks for hidden patterns that may exist in unlabelled data. It groups the data into disjointed clusters, ensuring that the data in each cluster is similar, yet different to the other data clusters. Clustering can be useful in image processing, information retrieval and pattern recognition. [4]

What is a k-Means Algorithm?

The k-means algorithm is a simple unsupervised clustering algorithm capable of clustering clear clusters of data very quickly and efficiently, often in just a few iterations. [5] As can be seen in figure 1 there are three easily identifiable clusters.

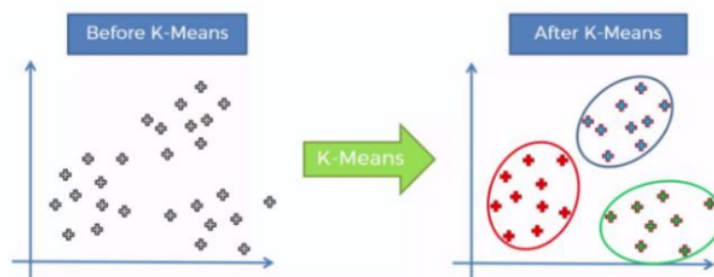


Figure 12 - Image of Unclustered Data, Clustered with K-means. [6]

k-Means is a popular algorithm due to its simplicity and general speed. It partitions the initial data into the required number of clusters, known as “k”. Each cluster has a centroid, given an initial starting value, the centroid value changes depending on the computed squared distances between the data point and the centroid. The smaller the sum of the squared distances the more accurate the clustering is considered. [4]

There are three common and widely used algorithms, they are [7]:

- **Lloyd & Forgy Algorithm**

1. Lloyd Algorithm (1957, published 1982)
2. Forgy Algorithm (1965)

Both Lloyd and Forgy Algorithms are batch centroid models, the main difference is that Lloyd’s algorithm considers the data distribution to be discrete, whereas Forgy’s algorithm considers the data distribution to be continuous.

A centroid is the geometric centre of the mean.

Batch algorithms are algorithms that see the transformative step being applied to all cases at the same time. They are best suited to large datasets to offset the computational expense of processing.

Steps for Lloyd and Forgy

1. Determine number of clusters to be used (referred to as k);
2. Determine the metric (see Appendix 1) to be used;
3. Determine the method for picking the initial centroids;
4. Assign initial centroids;
5. While metric (centroids, cases) > threshold
 - a. For $i \leq \text{number (nb) cases}$

Assign closest cluster according to the metric
 - b. Recalculate the centroids.

- **McQueen Algorithm (1967)**

The McQueen Algorithm is an iterative or incremental algorithm. It is very similar in how it is initialised; the most significant difference is the fact that the centroids are recalculated for every case change within a subspace or after each pass through all cases. It is considered more efficient as it updates the centroids more often and needs to perform a complete pass-through cases to converge on a solution for the optimum centroids.

Steps for McQueen [7]

1. Determine number of clusters to be used (referred to as k);
2. Determine the metric (see Appendix 1) to be used;
3. Determine the method for picking the initial centroids;
4. Assign initial centroids;
5. While metric (centroids, cases) > threshold
 - a. For $i \leq \text{cases}$
 - i. Assign case i to closest cluster according to the metric
 - ii. Recalculate centroids for the affected clusters
 - b. Recalculate the centroids.

- **Hartigan & Wong Algorithm**

The Hartigan and Wong Algorithm searches for partitions in the data space with locally optimal “within-cluster” Sum of Squared Errors (SSE). The initial centroids are assigned as they are in Lloyd/Forgy algorithm. After every updated step the SSE’s are compared, if the newly calculated SSE is lower than the previous one, then the newly assigned cluster is the new cluster. This comparison continues until the number of iterations required has been completed or the SSE comparison have the same values.

Steps for Hartigan & Wong:

1. Determine the number of clusters (referred to as k);
2. Determine the metric (see Appendix 1) to use;
3. Determine the method to pick initial centroids;
4. Assign initial centroids;
5. Assign cases to closest centroid;
6. Calculate centroids;
7. For $j \leq \text{nb clusters}$, if centroid j was updated last iteration

- a. Calculate Sum of Squared Errors (SSE) within cluster
- b. For $i \leq \text{nb cases in cluster}$
 - i. Compute SSE for cluster $k \neq j$ if case included
 - ii. If $\text{SSE cluster } k < \text{SSE cluster } j$, case change cluster.

Critically evaluate with improvements

There are several issues with k-means clustering. [7]

One of the first significant advantages is the fact that it is a quick to implement and can be a useful starting point when looking at clustering algorithms.

K-means clustering will always converge but may only find local minimum cluster solutions rather than global optimal clusters. This can be counteracted by doing multiple applications of the technique with different starting points, ensuring a more stable solution is obtained.

It is possible to create empty clusters if all centroids are moved from a cluster subspace at the same time.

The two final methods (McQueen and Hartigan/Wong) are sensitive to the order in which points are located – reaping different solutions depending on the order.

k-means has a bias to create equal sized cluster groups and so may not represent the group distributions effectively.

Consideration also needs to be made if there are outliers as it uses the mean, which is a descriptive statistic, where outliers are usually removed to normalise a dataset.

Algorithm	Advantage	Disadvantage
Lloyd	For large datasets	Slower convergence
	Discrete data distribution	Possible to create empty clusters
	Optimise total sum of squares	
Forgy	For large datasets	Slower convergence
	Continuous data distribution	Possible to create empty clusters
	Optimise total sum of squares	
McQueen	Fast initial convergence	Need to store the two nearest-cluster computations for each case
	Optimise total sum of squares	Sensitive to the order the algorithm is applied to the cases
Hartigan & Wong	Fast initial convergence	Need to store the two nearest-cluster computations for each case
	Optimise total sum of squares	Sensitive to the order the algorithm is applied to the cases

Table combining the advantages and disadvantages [7]

There are other algorithms that could be considered when working with clustering and include:
[7]

- Fuzzy k-mean clustering (Bezdek, 1981)
- Self-Organising Maps (Kohonen, 1982)
- Gaussian-expectation maximisation (GEM) (Dempster, Laird & Rubin, 1977)

Task 3

Using Excel to perform the k-means clustering seems to be a logical option due the dataset only having 10 points with 2 centroid points to use, and the ability to map the points onto a scatterplot.

Using the mean method with Hartigan & Wong k-means algorithm:

Step 1 – determine the number of k clusters to be used

Number of clusters (k) = 2

Step 2 – chosen metric is Euclidian metric

Step 3 – determine the position of the centroids (Student ID: 1808257/1)

Random Point 1 = (4.69, 2.92)

Random Point 2 = (1.55, 0.46)

Step 4 – assign the points to a centroid

Data points to be assigned:

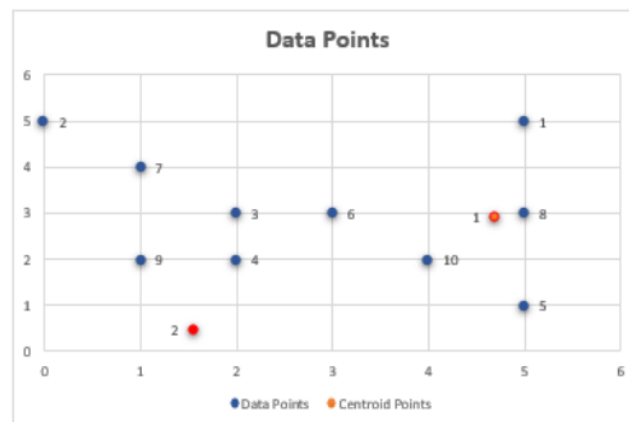
x	5	0	2	2	5	3	1	5	1	4
y	5	5	3	2	1	3	4	3	2	2

Mapping of the points and the centroid:

Initial Data Set and Mapping

Data Points		
Point	x	y
1	5	5
2	0	5
3	2	3
4	2	2
5	5	1
6	3	3
7	1	4
8	5	3
9	1	2
10	4	2

Centroid Points		
Cluster	x	y
1	4.69	2.92
2	1.55	0.46



Step 5 & 6 – Assign and calculate centroids:

Iteration 1

Data Points		
Point	x	y
1	5	5
2	0	5
3	2	3
4	2	2
5	5	1
6	3	3
7	1	4
8	5	3
9	1	2
10	4	2

Distance Squared	Cluster 1	Cluster 2	New Cluster
4.4225	4.4225	32.5141	1
23.0141	26.3225	23.0141	2
6.6541	7.2425	6.6541	2
2.5741	8.0825	2.5741	2
3.7825	3.7825	12.1941	1
2.8625	2.8625	8.5541	1
12.8341	14.7825	12.8341	2
0.1025	0.1025	18.3541	1
2.6741	14.4625	2.6741	2
1.3225	1.3225	8.3741	1

Centroid Points		
Cluster	x	y
1	4.69	2.92
2	1.55	0.46

SSE 60.243

Converged FALSE

Step 7 – Calculate the centroids of the newly formed cluster*Second Iteration:*

Iteration 2

Data Points		
Point	x	y
1	5	5
2	0	5
3	2	3
4	2	2
5	5	1
6	3	3
7	1	4
8	5	3
9	1	2
10	4	2

Cluster Group	Distance Squared	Cluster 1	Cluster 2	New Cluster
Cluster				
1	5.2	5.2	17.68	1
2	4.68	24.2	4.68	2
2	0.68	5.8	0.68	2
2	2.08	6.4	2.08	2
1	3.6	3.6	19.28	1
1	2	2	3.28	1
2	0.68	13	0.68	2
1	0.4	0.4	14.48	1
2	1.48	12.2	1.48	2
1	0.8	0.8	9.28	1

Centroid Points		
Cluster	x	y
1	4.4	2.8
2	1.2	3.2

SSE 21.6

Converged TRUE

Repeat steps 3 and 4 until the centroids remain the same

Third iteration proved they remained the same.

Third Iteration:

Iteration 3

Data Points		
Point	x	y
1	5	5
2	0	5
3	2	3
4	2	2
5	5	1
6	3	3
7	1	4
8	5	3
9	1	2
10	4	2

Cluster Group	Distance Squared	Cluster 1	Cluster 2	New Cluster
Cluster				
1	5.2	5.2	17.68	1
2	4.68	24.2	4.68	2
2	0.68	5.8	0.68	2
2	2.08	6.4	2.08	2
1	3.6	3.6	19.28	1
1	2	2	3.28	1
2	0.68	13	0.68	2
1	0.4	0.4	14.48	1
2	1.48	12.2	1.48	2
1	0.8	0.8	9.28	1

Centroid Points		
Cluster	x	y
1	4.4	2.8
2	1.2	3.2

SSE 21.6

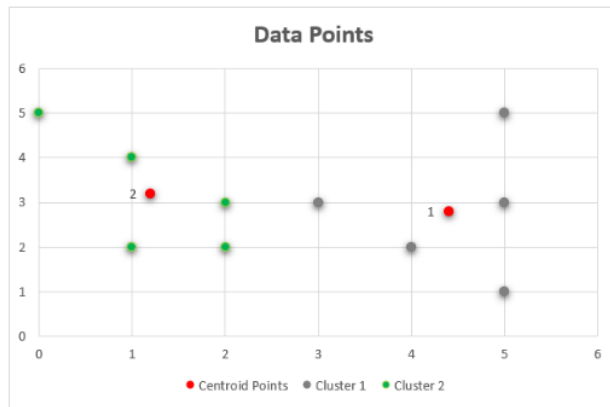
Converged TRUE

Mapping of the points and the centroid:

Final Data Set and Mapping

Data Points			
Point	x	y	Cluster
1	5	5	1
2	0	5	2
3	2	3	2
4	2	2	2
5	5	1	1
6	3	3	1
7	1	4	2
8	5	3	1
9	1	2	2
10	4	2	1

Centroid Points		
Cluster	x	y
1	4.4	2.8
2	1.2	3.2



References

- [1] P. R. S. Norvig, *Artificial Intelligence: a Modern Approach*, 3rd ed. Gl. Pearson, 2016.
- [2] "Stack in C++ STL - GeeksforGeeks." [Online]. Available: <https://www.geeksforgeeks.org/stack-in-cpp-stl/>. [Accessed: 04-Jan-2022].
- [3] "C++ Struct With Example." [Online]. Available: <https://www.guru99.com/cpp-structures.html>. [Accessed: 07-Jan-2022].
- [4] K. R. Žalik, "An efficient k'-means clustering algorithm," *Pattern Recognit. Lett.*, vol. 29, no. 9, pp. 1385–1391, Jul. 2008.
- [5] A. Géron, "Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems," *O'Reilly Media*, p. 851, 2019.
- [6] "K Means Clustering. In the previous story we understood... | by Ayush Kalla | DataDrivenInvestor." [Online]. Available: <https://medium.datadriveninvestor.com/k-means-clustering-4a700d4a4720>. [Accessed: 31-Dec-2021].
- [7] L. Morissette and S. Chartier, "The k-means clustering technique: General considerations and implementation in Mathematica," *Tutor. Quant. Methods Psychol.*, vol. 9, no. 1, pp. 15–24, 2013.

Appendices

Appendix 1

Where the metric is one of the following:

- Euclidian distance,
- Squared Euclidian distance,
- Manhattan distance,
- Maximum distance,
- Mahalanobis distance
- Cosine similarity

DFS Assignment Report 1808257

GRADEMARK REPORT

FINAL GRADE

65 / 100

GENERAL COMMENTS

Instructor

Code works, demo provided. Essential requirements met. Further work could include resolving the lack of backtracking from dead-ends on the animation and removing these from the path. Also would be nice to see Dijkstra's or A* to find shortest path once all signs were found by dfs.

Task 2.

A good review and analysis of k-means. Refs appropriate. More perhaps on the detail of potential methods used in selecting k. Also a further review and analysis of distance measures would help.

Task 3

Method used MSE. ok but please review alternatives and include reasons for choosing MSE.

Cals correct for this method.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12



Comment 1

Some problem here.

I've looked at your code so no problem - best check next time.

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

ISOBEL SUTCLIFFE

0

RESEARCH-INFORMED LITERATURE

KNOWLEDGE AND UNDERSTANDING

ANALYSIS

PRACTICAL APPLICATION AND DEP.

SKILLS FOR PROFESSIONAL PRAC.