DATA STRUCTURES AND ALGORITHMS

ASSIGNMENT 4

APNATVA SINGH RAWAT
21354929

Approach for Task 3:

1. Global variables used are
    a. MAX_DISTANCE - random number for initialisation of Dijsktra's Algorithm
    b. MAX_BUFFER - maximum buffer while reading a field from any csv
    c. MAX_VERTICES - maximum number of stops to initialise variables
    d. int graph - adjacency matrix for the undirected graph.
    e. int numV - to store the last index + 1 where a vertice has been stored.
    f. vertice arr - an array of struct type which will store the information we read from the vertices.csv file.
2. Load vertices from the file into a struct using the parser provided in Assignment 0 solution. A similar method of storing it using a global struct array (arr). It stores each stop such that the index number of the array where it is stored matches the stop number.
    a. Functions used for this part - load_vertices, verticeUtility, next.
3. Load edges to temporary variables and add edges using the addedge function. This function helps me create an adjacency matrix for the undirected graph.
    a. Functions used for this part - load_edges, edgeUtility, next.
4. To print the shortest path I have used Dijsktra's Algorithm. The function from task 2 was renamed as shortest_path. This would calculate the minimum distance. But for task 3 we had to print the path, hence an extra array named 'parent' was used to store the parent for each node. This would be updated whenever a shorter path was found to the node. This part was very much inspired from GeeksForGeeks.
    a. Functions used:
        i. shortest_path: finds the minimum path
        ii. minDistance: finds the next closest node to make it permanent.
        iii. printPath: recursive function to print the path. Each node will have only one parent while one node can have many children. Hence we will have to backtrack from the last node and then go towards the first node.The print statement is after the function call so that it would be printed the reverse order (i.e. from first to last stop).
    b.
5. free_memory is an empty function since I did not allocate any memory using malloc, hence no need to free any memory.

Outputs for Task 1, Task 2 and Task 3 are attached below.

14 / 14 Autograding Total

C Compilation part 1

2 / 2 C Testing part 1

Student STDOUT.txt

```
DFS: A  B  C  D  F  E
BFS  A  B  D  E  C  F
```

Expected STDOUT.txt

```
DFS: A  B  C  D  F  E
BFS  A  B  D  E  C  F
```

1 / 1 Using Valgrind to check for memory leaks

Student Standard Error (STDERR)

```
==2563990== Memcheck, a memory error detector
==2563990== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==2563990== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==2563990== Command: ./p1.out
==2563990==
==2563990==
==2563990== HEAP SUMMARY:
==2563990==     in use at exit: 0 bytes in 0 blocks
==2563990==   total heap usage: 17 allocs, 17 frees, 4,480 bytes allocated
==2563990==
==2563990== All heap blocks were freed -- no leaks are possible
==2563990==
==2563990== For lists of detected and suppressed errors, rerun with: -s
==2563990== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

C Compilation part 1 - test 2

1 / 1 C Testing part 1 - test 2

Student STDOUT.txt

```
DFS: A  B  D  F  C  E
BFS  A  B  D  E  F  C
```

Expected STDOUT.txt

```
DFS: A  B  D  F  C  E
BFS  A  B  D  E  F  C
```

C Compilation part 1 - test 3

1 / 1 C Testing part 1 - test 3

Student STDOUT.txt

```
DFS: A  B  C  D  F  E  G
BFS  A  B  D  E  C  F  G
```

Expected STDOUT.txt

```
DFS: A  B  C  D  F  E  G
BFS  A  B  D  E  C  F  G
```

C Compilation part 2

2 / 2 C Testing part 2!

Student STDOUT.txt

```
A B C G E D F
The length of the shortest path between A and A is 0
The length of the shortest path between A and B is 1
The length of the shortest path between A and C is 2
The length of the shortest path between A and D is 7
The length of the shortest path between A and E is 5
The length of the shortest path between A and F is 7
The length of the shortest path between A and G is 3
```

Expected STDOUT.txt

```
A B C G E D F
The length of the shortest path between A and A is 0
The length of the shortest path between A and B is 1
The length of the shortest path between A and C is 2
The length of the shortest path between A and D is 7
The length of the shortest path between A and E is 5
The length of the shortest path between A and F is 7
The length of the shortest path between A and G is 3
```

1 / 1 Using Valgrind to check for memory leaks

Student Standard Error (STDERR)

```
==2564212== Memcheck, a memory error detector
==2564212== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==2564212== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==2564212== Command: ./p2.out
==2564212==
==2564212==
==2564212== HEAP SUMMARY:
==2564212==     in use at exit: 0 bytes in 0 blocks
==2564212==   total heap usage: 8 allocs, 8 frees, 4,208 bytes allocated
==2564212==
==2564212== All heap blocks were freed -- no leaks are possible
==2564212==
==2564212== For lists of detected and suppressed errors, rerun with: -s
==2564212== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

C Compilation part 2 - test 2

1 / 1 C Testing part 2 - test 2

Student STDOUT.txt

```
J G I H F E D A B C
The length of the shortest path between J and A is 580
The length of the shortest path between J and B is 590
The length of the shortest path between J and C is 600
The length of the shortest path between J and D is 480
The length of the shortest path between J and E is 390
The length of the shortest path between J and F is 360
The length of the shortest path between J and G is 250
The length of the shortest path between J and H is 300
The length of the shortest path between J and I is 280
The length of the shortest path between J and J is 0
```

Expected STDOUT.txt

```
J G I H F E D A B C
The length of the shortest path between J and A is 580
The length of the shortest path between J and B is 590
The length of the shortest path between J and C is 600
The length of the shortest path between J and D is 480
The length of the shortest path between J and E is 390
The length of the shortest path between J and F is 360
The length of the shortest path between J and G is 250
The length of the shortest path between J and H is 300
The length of the shortest path between J and I is 280
The length of the shortest path between J and J is 0
```

C Compilation part 3

2 / 2 C Testing part 3!

Student STDOUT.txt

```
Loaded 4806 vertices
Loaded 6179 edges
Please enter stating bus stop >        Please enter destination bus stop >
497 Amiens Street
515 Amiens Street
516 North Strand Rd
4384 North Strand Rd
519 North Strand Rd
521 Annesley Bridge
522 Marino Mart
523 Marino Mart
669 Malahide Road
670 Malahide Road
671 Malahide Road
672 Malahide Road
4382 Malahide Road
1185 Collins Ave
1186 Collins Ave
1187 Collins Ave
1188 Collins Ave
1189 Collins Ave
216 Beaumont Road
217 Beaumont Road
242 Beaumont Road
243 Beaumont Road
253 Beaumont Hospital
```

Expected STDOUT.txt

```
Loaded 4806 vertices
Loaded 6179 edges
Please enter stating bus stop >        Please enter destination bus stop >
497 Amiens Street
515 Amiens Street
516 North Strand Rd
4384 North Strand Rd
519 North Strand Rd
521 Annesley Bridge
522 Marino Mart
523 Marino Mart
669 Malahide Road
670 Malahide Road
671 Malahide Road
672 Malahide Road
4382 Malahide Road
1185 Collins Ave
1186 Collins Ave
1187 Collins Ave
1188 Collins Ave
1189 Collins Ave
216 Beaumont Road
217 Beaumont Road
242 Beaumont Road
243 Beaumont Road
253 Beaumont Hospital
```

Student Execution Logfile

```
Child exited with status = 1
```

1 / 1 Using Valgrind to check for memory leaks

Student Standard Error (STDERR)

```
==2564451== Memcheck, a memory error detector
==2564451== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==2564451== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==2564451== Command: ./p3.out
==2564451==
==2564451==
==2564451== HEAP SUMMARY:
==2564451==     in use at exit: 0 bytes in 0 blocks
==2564451==   total heap usage: 1 allocs, 1 frees, 4,096 bytes allocated
==2564451==
==2564451== All heap blocks were freed -- no leaks are possible
==2564451==
==2564451== For lists of detected and suppressed errors, rerun with: -s
==2564451== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Student Execution Logfile

```
Child exited with status = 1
```

1 / 1 C Testing part 3 - test 2!

Student STDOUT.txt

```
Loaded 4806 vertices
Loaded 6179 edges
Please enter stating bus stop >        Please enter destination bus stop >
747 Kildare Street
748 Merrion Row
2905 Merrion Sq West
494 Clare Street
495 Westland Row
496 Beresford Place
497 Amiens Street
515 Amiens Street
516 North Strand Rd
4384 North Strand Rd
519 North Strand Rd
521 Annesley Bridge
522 Marino Mart
523 Marino Mart
669 Malahide Road
670 Malahide Road
671 Malahide Road
672 Malahide Road
4382 Malahide Road
1185 Collins Ave
1186 Collins Ave
1187 Collins Ave
1188 Collins Ave
1189 Collins Ave
216 Beaumont Road
217 Beaumont Road
218 Shantalla Road
219 Shantalla Road
220 Swords Road
1622 Swords Road
1623 Swords Road
1624 Swords Road
1625 Swords Road
1626 Swords Road
1627 Swords Road
1628 Swords Road
1629 Swords Road
1630 Swords Road
7348 Dublin Airport
3663 Dublin Airport
```

Expected STDOUT.txt

```
Loaded 4806 vertices
Loaded 6179 edges
Please enter stating bus stop >        Please enter destination bus stop >
747 Kildare Street
748 Merrion Row
2905 Merrion Sq West
494 Clare Street
495 Westland Row
496 Beresford Place
497 Amiens Street
515 Amiens Street
516 North Strand Rd
4384 North Strand Rd
519 North Strand Rd
521 Annesley Bridge
522 Marino Mart
523 Marino Mart
669 Malahide Road
670 Malahide Road
671 Malahide Road
672 Malahide Road
4382 Malahide Road
1185 Collins Ave
1186 Collins Ave
1187 Collins Ave
1188 Collins Ave
1189 Collins Ave
216 Beaumont Road
217 Beaumont Road
218 Shantalla Road
219 Shantalla Road
220 Swords Road
1622 Swords Road
1623 Swords Road
1624 Swords Road
1625 Swords Road
1626 Swords Road
1627 Swords Road
1628 Swords Road
1629 Swords Road
1630 Swords Road
7348 Dublin Airport
3663 Dublin Airport
```

Student Execution Logfile

```
Child exited with status = 1
```

1 / 1 C Testing part 3 - test 3!

Student STDOUT.txt

```
Loaded 4806 vertices
Loaded 6179 edges
Please enter stating bus stop >        Please enter destination bus stop >
3235 Pearse Street
3236 Sallynoggin Road
3343 Sallynoggin Road
3248 Sallynoggin Road
7056 Rochestown Ave
4731 Rochestown Ave
3249 Pottery Rd
7667 Barnhill Rd
7652 Killiney Hill Rd
```

Expected STDOUT.txt

```
Loaded 4806 vertices
Loaded 6179 edges
Please enter stating bus stop >        Please enter destination bus stop >
3235 Pearse Street
3236 Sallynoggin Road
3343 Sallynoggin Road
3248 Sallynoggin Road
7056 Rochestown Ave
4731 Rochestown Ave
3249 Pottery Rd
7667 Barnhill Rd
7652 Killiney Hill Rd
```