Nayeel Imtiaz
CSE 13s
10/21/21

# Project 4 - The Perambulations of Denver Long

Summary: The main usage of this program is to find the shortest path for both directed and undirected graphs. Graphs are represented by adjacency matrices in this program. The final executable will include a graph abstract data type, a stack abstract data type, a shortest path finding algorithm, and the main file that will parse user arguments. Graph.c and graph.h will contain code for the graph structure. Stack.c and stack.h will contain code for the stack structure.

Pseudocode:

Tsp.c -

```
Parse through input
While there are inputs to be read:
    Mark which command arguments are inputted.
Run tests that are flagged.
Provide statistics if -v is flagged.
Make the graph undirected if -u is flagged.
Specify in file to take input from if -i is flagged.
Specify output file to print to if -o is flagged.
Provide instructions if -h is flagged.
```

graph.c -

```
i = 0
j = 0
k = 0
While unvisited:
    Perform dps algorithm
    Add k by 1.
    Add counter by 1.
    Add current point to visited.
Return the value
```

stack.c

```
Length = 0
Dynamically allocate enough space for stack using calloc.
List = [ ]
If pop:
    Remove last element from list if you run the pop function.
    Length -= 1
If push:
    Append to the end if you run the push function.
    Length += 1
```