Minimum Spanning Tree

Implementation of Prim's Algorithm

This document provides detailed documentation for the implementation of Prim's algorithm for finding the Minimum Spanning Tree of an undirected graph.

Functions Implemented

Binary Heap Operations

Min-Heapify

This function maintains the min-heap property by recursively ensuring that a node is smaller than its children.

```
/**
 * minHeapify(h, i):
 *   Restore heap property at index i by sifting down.
 */
void minHeapify(MinHeap *h, int i);
```

Extract-Min

Extracts the element with the highest priority (minimum value) from the heap.

```
/**
 * extractMin(h):
 *   Remove & return the root (smallest) node.
 *   Marks its pos[] = 0 so it's no longer in the heap.
 */
HeapNode extractMin(MinHeap *h);
```

Decrease-Key

Decreases the value of a node's key and updates the heap structure accordingly.

```
/**
 * decreaseKey(h, v, newKey):
 *   Lower the key of vertex v and bubble it up.
 */
void decreaseKey(MinHeap *h, int v, int newKey);
```

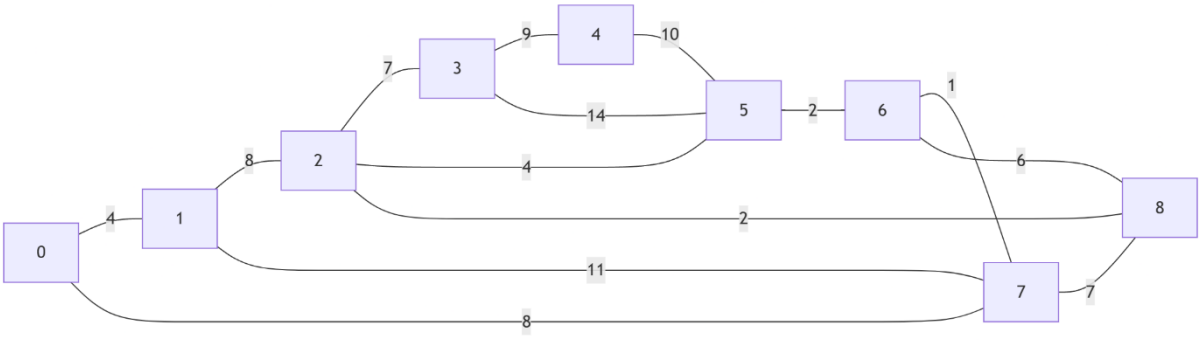
MST-Prim Algorithm

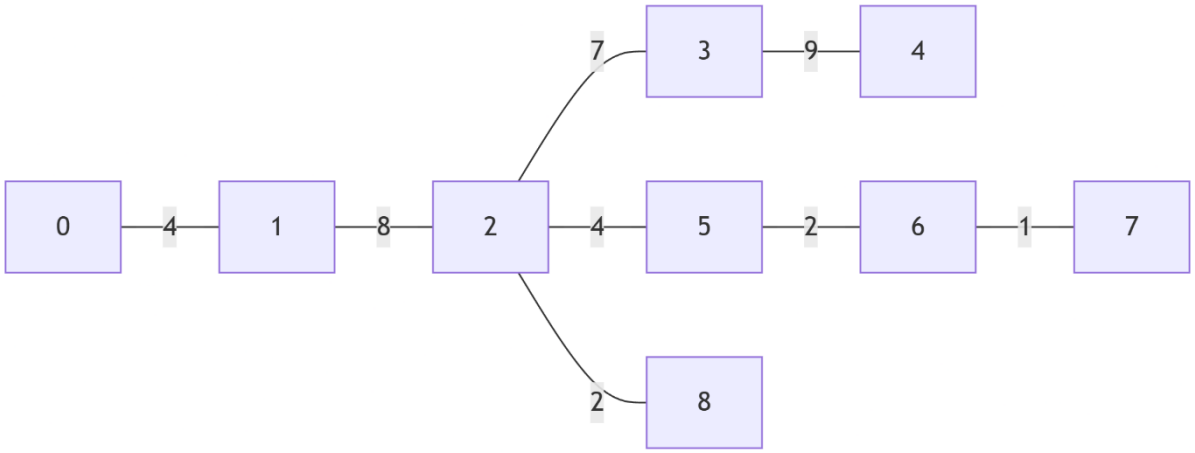
```
/**
 * primMST(adj, n, src, parent):
 *   Runs Prim's algorithm with a min-heap in O(m log n).
 *   Fills parent[v] = predecessor of v in the MST.
 */
void primMST(Edge **adj, int n, int src, int *parent);
```

Test Results and Screenshots

1. A 9-vertex graph with 14 edges

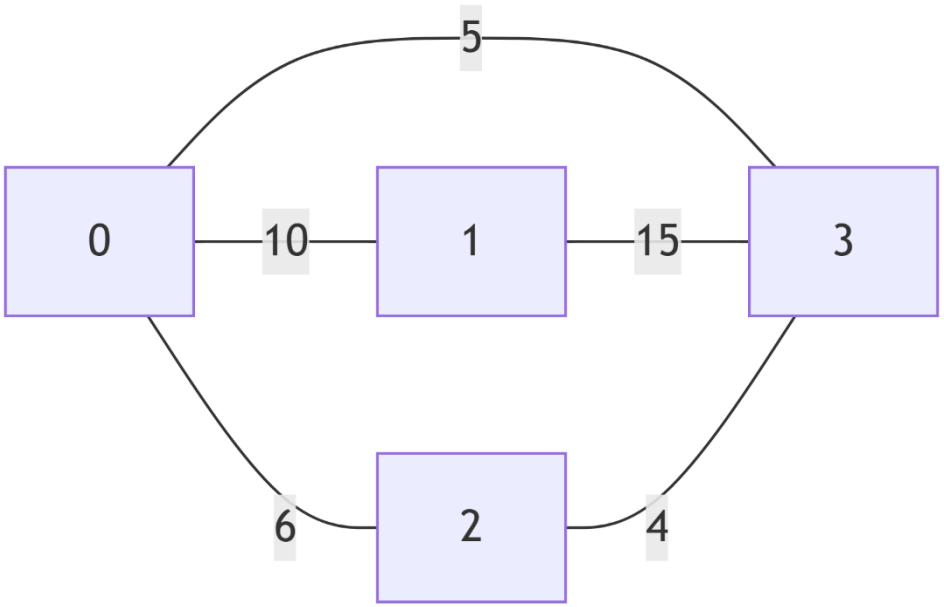
```
root@DESKTOP-NicoSpyker:/mnt/c/Users/Spijkerman/Repositories/minimum_spanning_tree# cat input_mst.txt
9       14      0
0       1       4
0       7       8
1       2       8
1       7       11
2       3       7
2       5       4
2       8       2
3       4       9
3       5       14
4       5       10
5       6       2
6       7       1
6       8       6
7       8       7
root@DESKTOP-NicoSpyker:/mnt/c/Users/Spijkerman/Repositories/minimum_spanning_tree# make run
gcc -W -Wall -Wextra -Werror -O2   -c -o src/main.o src/main.c
gcc ./src/main.o -o mst
./mst input_mst.txt output_mst.txt
root@DESKTOP-NicoSpyker:/mnt/c/Users/Spijkerman/Repositories/minimum_spanning_tree# cat output_mst.txt
0       NIL
1       0
2       1
3       2
4       3
5       2
6       5
7       6
8       2
```

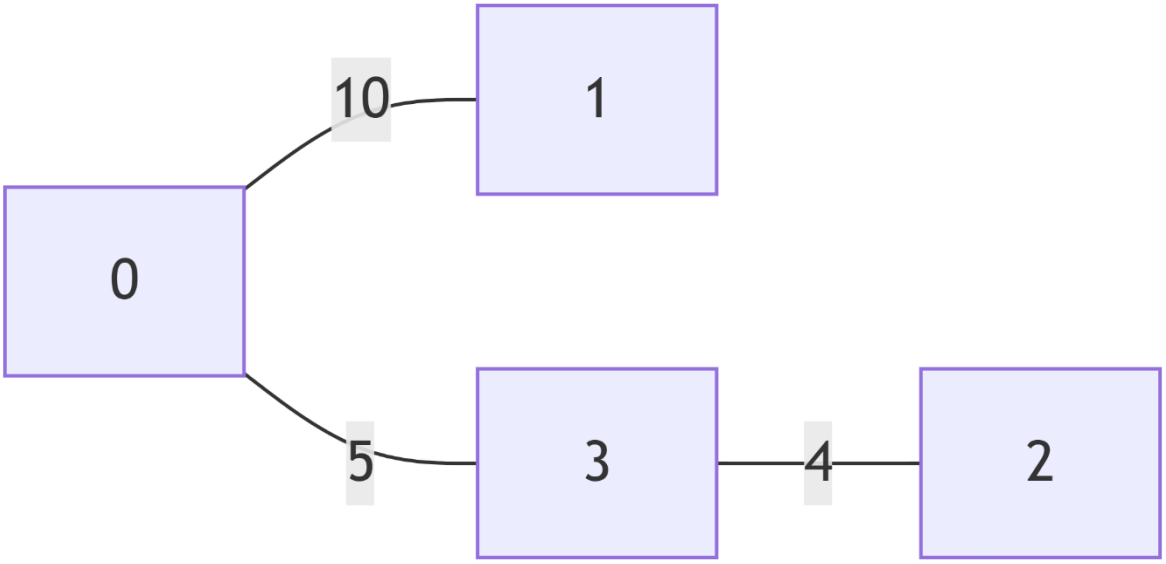

0
1
2
3
4
5
6
7
8
4
8
7
9
10
14
4
2
11
8
1
6
2
7


0
1
2
3
4
5
6
7
8
4
8
7
9
4
2
2
1

2. A small 4-vertex graph with 5 edges

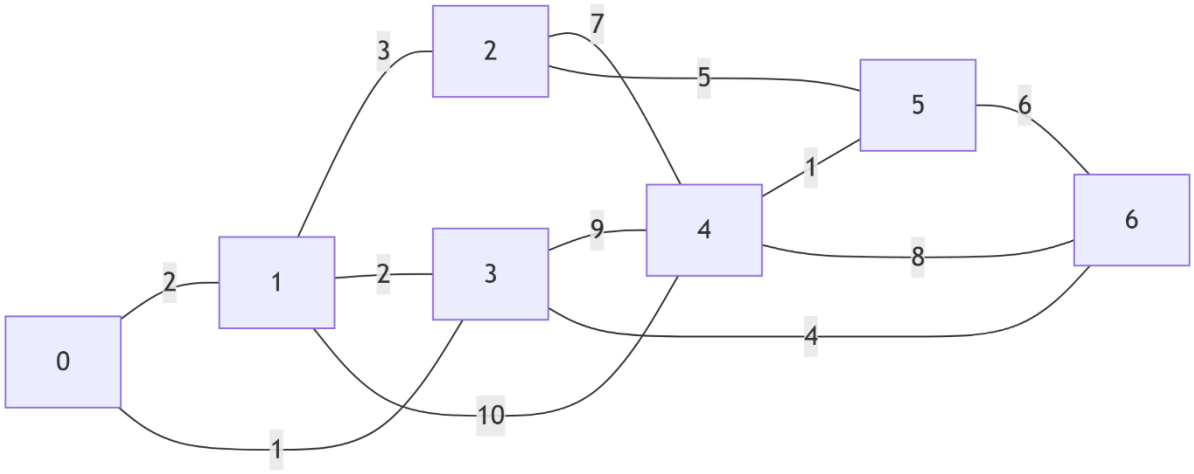
```
root@DESKTOP-NicoSpyker:/mnt/c/Users/Spijkerman/Repositories/minimum_spanning_tree# cat input_mst.txt
4       5       0
0       1       10
0       2       6
0       3       5
1       3       15
2       3       4
root@DESKTOP-NicoSpyker:/mnt/c/Users/Spijkerman/Repositories/minimum_spanning_tree# make run
gcc -W -Wall -Wextra -Werror -O2    -c -o src/main.o src/main.c
gcc ./src/main.o -o mst
./mst input_mst.txt output_mst.txt
root@DESKTOP-NicoSpyker:/mnt/c/Users/Spijkerman/Repositories/minimum_spanning_tree# cat output_mst.txt
0       NIL
1       0
2       3
3       0
```

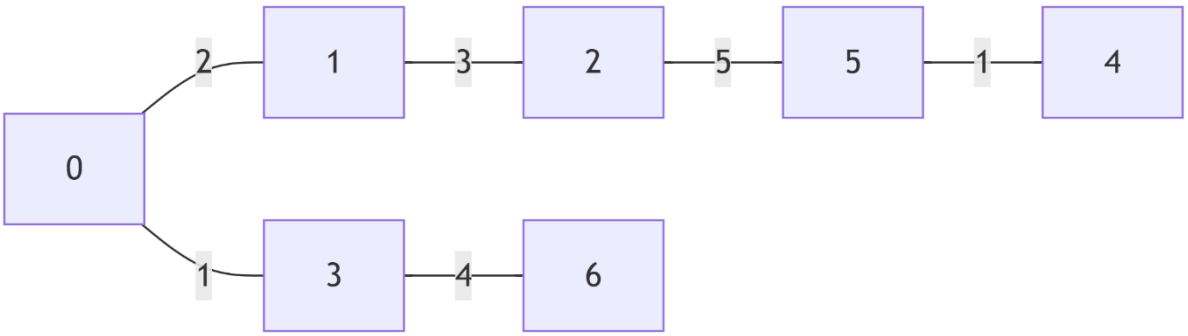




3. A 7-vertex graph with 12 edges

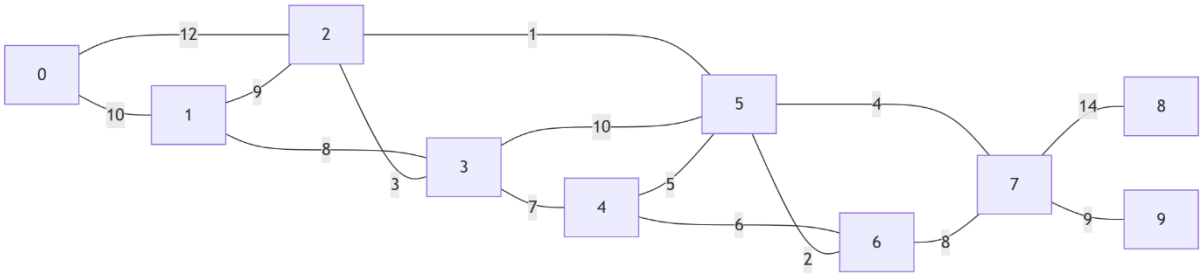
```
root@DESKTOP-NicoSpyker:/mnt/c/Users/Spijkerman/Repositories/minimum_spanning_tree# cat input_mst.txt
7       12      0
0       1       2
0       3       1
1       2       3
1       3       2
1       4       10
2       4       7
2       5       5
3       4       9
3       6       4
4       5       1
4       6       8
5       6       6
root@DESKTOP-NicoSpyker:/mnt/c/Users/Spijkerman/Repositories/minimum_spanning_tree# make run
gcc -W -Wall -Wextra -Werror -O2   -c -o src/main.o src/main.c
gcc ./src/main.o -o mst
./mst input_mst.txt output_mst.txt
root@DESKTOP-NicoSpyker:/mnt/c/Users/Spijkerman/Repositories/minimum_spanning_tree# cat output_mst.txt
0       NIL
1       0
2       1
3       0
4       5
5       2
6       3
```

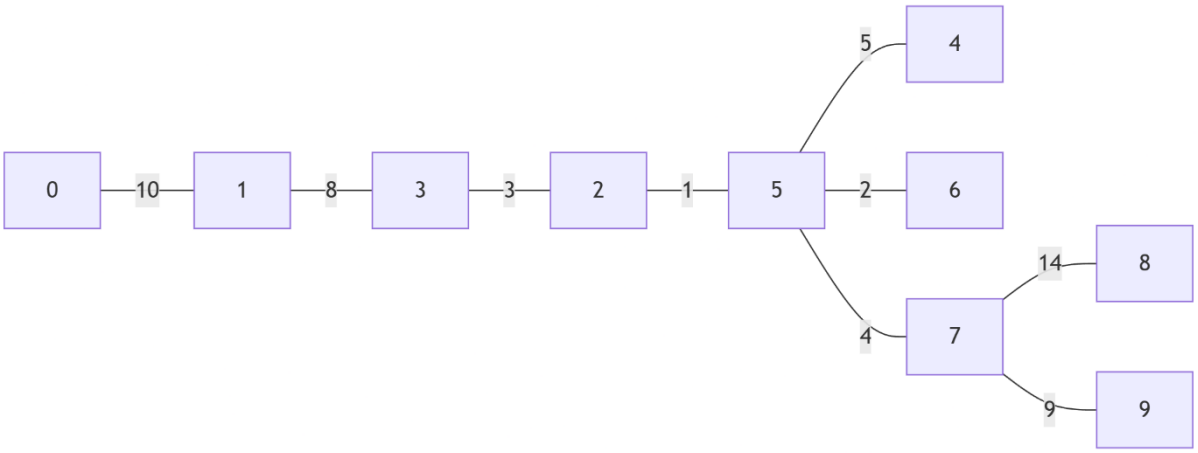

0
1
2
3
4
5
6
2
3
7
5
6
1
9
8
4
10
1


0
1
2
3
4
5
6
2
3
5
1
1
4
6

4. A 10-vertex graph with 15 edges

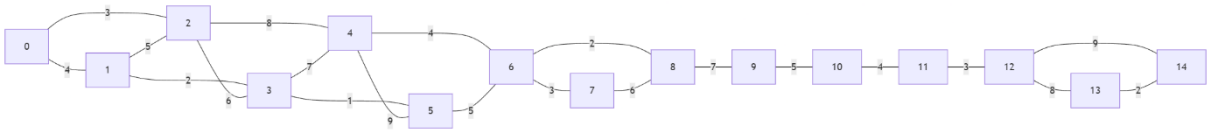
```
root@DESKTOP-NicoSpyker:/mnt/c/Users/Spijkerman/Repositories/minimum_spanning_tree# cat input_mst.txt
10      15      0
0       1       10
0       2       12
1       2       9
1       3       8
2       3       3
2       5       1
3       4       7
3       5       10
4       5       5
4       6       6
5       6       2
5       7       4
6       7       8
7       8       14
7       9       9
root@DESKTOP-NicoSpyker:/mnt/c/Users/Spijkerman/Repositories/minimum_spanning_tree# make run
gcc -W -Wall -Wextra -Werror -O2   -c -o src/main.o src/main.c
gcc ./src/main.o -o mst
./mst input_mst.txt output_mst.txt
root@DESKTOP-NicoSpyker:/mnt/c/Users/Spijkerman/Repositories/minimum_spanning_tree# cat output_mst.txt
0       NIL
1       0
2       3
3       1
4       5
5       2
6       5
7       5
8       7
9       7
```

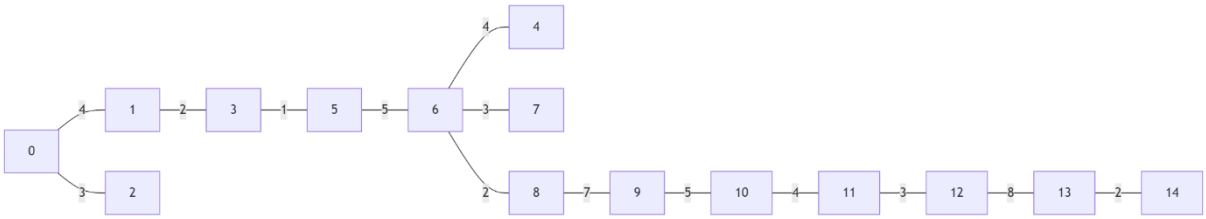

0
1
2
3
4
5
6
7
8
9
12
10
9
1
8
3
10
5
7
6
2
4
8
14
9


0
1
3
2
5
4
6
7
8
9
10
8
3
1
5
2
4
14
9

5. A 15-vertex graph with 21 edges

```
root@DESKTOP-NicoSpyker:/mnt/c/Users/Spijkerman/Repositories/minimum_spanning_tree# cat input_mst.txt
15      21      0
0       1       4
0       2       3
1       2       5
1       3       2
2       3       6
2       4       8
3       4       7
3       5       1
4       5       9
4       6       4
5       6       5
6       7       3
6       8       2
7       8       6
8       9       7
9       10      5
10      11      4
11      12      3
12      13      8
12      14      9
13      14      2
root@DESKTOP-NicoSpyker:/mnt/c/Users/Spijkerman/Repositories/minimum_spanning_tree# make run
gcc -W -Wall -Wextra -Werror -O2   -c -o src/main.o src/main.c
gcc ./src/main.o -o mst
./mst input_mst.txt output_mst.txt
root@DESKTOP-NicoSpyker:/mnt/c/Users/Spijkerman/Repositories/minimum_spanning_tree# cat output_mst.txt
0       NIL
1       0
2       0
3       1
4       6
5       3
6       5
7       6
8       6
9       8
10      9
11      10
12      11
13      12
14      13
```


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14