CSCE 4133
Algorithms
Programming Assignment 4 Report

Name:
Clayton Warstler – 010971514 – cjwarstl@uark.edu

Date:
11 November 2023

Academic Integrity Statement: I pledge that I have neither given nor received unauthorized help on this programming assignment.

Problem Statement:

For this project, I was tasked with implementing a Minimum Spanning Tree as if the university wanted to install cables between every building on campus. To implement this, we were to use the algorithm known as Kruskal's algorithm. This algorithm allows us to choose a set of connections, or edges, and the cost of each connection, or the weight of the edge. The only thing we had to implement was the sorting algorithms, quicksort, mergesort, and heapsort.

Implementation:

I started by downloading the homework file provided on Blackboard. The pseudocode provided was not as filled out compared to the last assignment, instead, it was all on our notes. I needed to add a couple extra functions for my implementations of the three sorting algorithms. I added a swap function, a partition function, a merge function, and a heapify function, all of which are shown in more detail below.

```
template<class T>
void swap(T &a, T &b);

template<class T>
int partition(std::vector<T> &array, int l, int r);

template<class T>
void merge(std::vector<T> &array, int l, int m, int r);

template<class T>
void heapify(std::vector<T> &array, int n, int i);
```

Testing:

All the makefile instructions as well as the compilation commands were given to us in the review and instructions. We also were given the required test cases to test all the functions. This means that the only testing I had to do was run the program with the given commands. This streamlined the testing process, meaning that I could "test" the program in seconds. I am extremely grateful for this.

The results of running the makefile are below:

qsort:

```
C:\Users\clayw\source\repos\WarstlerC_Algorithms_Assignment4\HW4\code>mingw32-make qsort
```

```
C:/mingw64/bin/g++.exe -I./include/ -std=c++11 -DOPENCV=1 -IC:/opencv/build/install/include -std=c++11 src/algorithms/mst.cpp src/graph.cpp src/main.cpp src/sort/qsort.cpp -LC:/opencv/build/install/x64/mingw/bin -lopencv_imgcodecs3413 -lopencv_core3413 -lopencv_highgui3413 -lopencv_imgproc3413 -o bin/qsort
./bin/qsort
Perform unit test on the sorting algorithm
You are using Quick Sort algorithm
Your sorting implementation is correct


Perform unit test on your implementation with graph
Minimum Spanning Tree:
Edge: 1 0. Cost: 1
Edge: 2 1. Cost: 2
Edge: 3 1. Cost: 3
Edge: 4 2. Cost: 4
Edge: 5 4. Cost: 6
Total Cost: 16


Minimum Spanning Tree:
Edge: 136 24. Cost: 20
Edge: 72 29. Cost: 21
Edge: 118 117. Cost: 22
Edge: 131 74. Cost: 23
Edge: 130 56. Cost: 23
Edge: 75 108. Cost: 24
Edge: 1 116. Cost: 24
Edge: 134 17. Cost: 25
Edge: 65 47. Cost: 26
Edge: 133 132. Cost: 27
Edge: 98 107. Cost: 28
Edge: 109 110. Cost: 28
Edge: 42 16. Cost: 29
Edge: 92 44. Cost: 29
Edge: 57 96. Cost: 29
Edge: 130 131. Cost: 29
Edge: 11 112. Cost: 30
Edge: 135 134. Cost: 31
Edge: 115 89. Cost: 31
Edge: 117 94. Cost: 32
Edge: 3 23. Cost: 32
Edge: 8 97. Cost: 33
Edge: 93 34. Cost: 33
Edge: 0 58. Cost: 33
Edge: 108 109. Cost: 34
Edge: 125 124. Cost: 35
Edge: 83 92. Cost: 37
Edge: 69 76. Cost: 38
Edge: 37 22. Cost: 38
Edge: 26 87. Cost: 39
```

Edge: 121 122. Cost: 39
Edge: 7 10. Cost: 40
Edge: 55 11. Cost: 41
Edge: 123 124. Cost: 41
Edge: 65 79. Cost: 42
Edge: 129 85. Cost: 42
Edge: 7 105. Cost: 43
Edge: 94 10. Cost: 43
Edge: 20 128. Cost: 43
Edge: 82 52. Cost: 44
Edge: 106 105. Cost: 44
Edge: 75 29. Cost: 44
Edge: 36 106. Cost: 44
Edge: 96 40. Cost: 46
Edge: 35 54. Cost: 46
Edge: 74 133. Cost: 47
Edge: 2 5. Cost: 47
Edge: 135 59. Cost: 48
Edge: 97 73. Cost: 48
Edge: 62 44. Cost: 48
Edge: 110 111. Cost: 48
Edge: 126 127. Cost: 48
Edge: 83 15. Cost: 50
Edge: 28 95. Cost: 51
Edge: 22 21. Cost: 51
Edge: 62 52. Cost: 52
Edge: 0 77. Cost: 53
Edge: 36 17. Cost: 54
Edge: 96 82. Cost: 54
Edge: 30 18. Cost: 54
Edge: 137 12. Cost: 55
Edge: 67 100. Cost: 56
Edge: 44 16. Cost: 57
Edge: 72 66. Cost: 57
Edge: 8 69. Cost: 57
Edge: 9 114. Cost: 57
Edge: 53 35. Cost: 59
Edge: 102 43. Cost: 59
Edge: 42 90. Cost: 60
Edge: 41 95. Cost: 61
Edge: 84 42. Cost: 64
Edge: 116 115. Cost: 64
Edge: 102 81. Cost: 64
Edge: 47 30. Cost: 67
Edge: 102 11. Cost: 67
Edge: 27 56. Cost: 68
Edge: 111 93. Cost: 68
Edge: 6 111. Cost: 69
Edge: 121 39. Cost: 69
Edge: 5 87. Cost: 69
Edge: 137 60. Cost: 70

Edge: 60 45. Cost: 70
Edge: 36 24. Cost: 71
Edge: 91 15. Cost: 74
Edge: 29 32. Cost: 76
Edge: 19 38. Cost: 77
Edge: 71 86. Cost: 77
Edge: 118 50. Cost: 77
Edge: 127 101. Cost: 79
Edge: 18 6. Cost: 80
Edge: 27 64. Cost: 80
Edge: 73 84. Cost: 83
Edge: 107 106. Cost: 84
Edge: 25 58. Cost: 84
Edge: 91 80. Cost: 86
Edge: 125 126. Cost: 86
Edge: 71 80. Cost: 87
Edge: 70 67. Cost: 87
Edge: 3 51. Cost: 88
Edge: 20 1. Cost: 89
Edge: 46 80. Cost: 90
Edge: 128 129. Cost: 90
Edge: 54 80. Cost: 91
Edge: 11 113. Cost: 98
Edge: 20 79. Cost: 98
Edge: 28 9. Cost: 100
Edge: 78 40. Cost: 101
Edge: 73 4. Cost: 102
Edge: 33 39. Cost: 102
Edge: 9 63. Cost: 103
Edge: 93 114. Cost: 103
Edge: 41 49. Cost: 105
Edge: 21 88. Cost: 106
Edge: 90 19. Cost: 107
Edge: 71 43. Cost: 112
Edge: 82 25. Cost: 112
Edge: 2 41. Cost: 113
Edge: 119 61. Cost: 115
Edge: 104 100. Cost: 116
Edge: 103 61. Cost: 116
Edge: 60 11. Cost: 122
Edge: 56 33. Cost: 122
Edge: 51 99. Cost: 125
Edge: 138 61. Cost: 130
Edge: 120 119. Cost: 130
Edge: 88 4. Cost: 131
Edge: 63 101. Cost: 133
Edge: 63 119. Cost: 137
Edge: 100 101. Cost: 139
Edge: 55 10. Cost: 147
Edge: 99 85. Cost: 154
Edge: 14 22. Cost: 158

Edge: 48 123. Cost: 166
Edge: 43 5. Cost: 286
Total Cost: 9290

msort:
C:\Users\clayw\source\repos\WarstlerC_Algorithms_Assignment4\HW4\code>mingw32-make msort
C:/mingw64/bin/g++.exe -I./include/ -std=c++11 -DOPENCV=1 -IC:/opencv/build/install/include -std=c++11 src/algorithms/mst.cpp src/graph.cpp src/main.cpp src/sort/msort.cpp -LC:/opencv/build/install/x64/mingw/bin -lopencv_imgcodecs3413 -lopencv_core3413 -lopencv_highgui3413 -lopencv_imgproc3413 -o bin/msort
./bin/msort
Perform unit test on the sorting algorithm
You are using Merge Sort algorithm
Your sorting implementation is correct

Perform unit test on your implementation with graph
Minimum Spanning Tree:
Edge: 0 1. Cost: 1
Edge: 1 2. Cost: 2
Edge: 1 3. Cost: 3
Edge: 2 4. Cost: 4
Edge: 4 5. Cost: 6
Total Cost: 16

Minimum Spanning Tree:
Edge: 24 136. Cost: 20
Edge: 29 72. Cost: 21
Edge: 117 118. Cost: 22
Edge: 56 130. Cost: 23
Edge: 74 131. Cost: 23
Edge: 1 116. Cost: 24
Edge: 75 108. Cost: 24
Edge: 17 134. Cost: 25
Edge: 47 65. Cost: 26
Edge: 132 133. Cost: 27
Edge: 98 107. Cost: 28
Edge: 109 110. Cost: 28
Edge: 16 42. Cost: 29
Edge: 44 92. Cost: 29
Edge: 57 96. Cost: 29
Edge: 130 131. Cost: 29
Edge: 11 112. Cost: 30
Edge: 89 115. Cost: 31
Edge: 134 135. Cost: 31
Edge: 3 23. Cost: 32
Edge: 94 117. Cost: 32
Edge: 0 58. Cost: 33
Edge: 8 97. Cost: 33
Edge: 34 93. Cost: 33

Edge: 108 109. Cost: 34
Edge: 124 125. Cost: 35
Edge: 83 92. Cost: 37
Edge: 22 37. Cost: 38
Edge: 69 76. Cost: 38
Edge: 26 87. Cost: 39
Edge: 121 122. Cost: 39
Edge: 7 10. Cost: 40
Edge: 11 55. Cost: 41
Edge: 123 124. Cost: 41
Edge: 65 79. Cost: 42
Edge: 85 129. Cost: 42
Edge: 7 105. Cost: 43
Edge: 10 94. Cost: 43
Edge: 20 128. Cost: 43
Edge: 29 75. Cost: 44
Edge: 36 106. Cost: 44
Edge: 52 82. Cost: 44
Edge: 105 106. Cost: 44
Edge: 35 54. Cost: 46
Edge: 40 96. Cost: 46
Edge: 2 5. Cost: 47
Edge: 74 133. Cost: 47
Edge: 44 62. Cost: 48
Edge: 59 135. Cost: 48
Edge: 73 97. Cost: 48
Edge: 110 111. Cost: 48
Edge: 126 127. Cost: 48
Edge: 15 83. Cost: 50
Edge: 21 22. Cost: 51
Edge: 28 95. Cost: 51
Edge: 52 62. Cost: 52
Edge: 0 77. Cost: 53
Edge: 17 36. Cost: 54
Edge: 18 30. Cost: 54
Edge: 82 96. Cost: 54
Edge: 12 137. Cost: 55
Edge: 67 100. Cost: 56
Edge: 8 69. Cost: 57
Edge: 9 114. Cost: 57
Edge: 16 44. Cost: 57
Edge: 66 72. Cost: 57
Edge: 35 53. Cost: 59
Edge: 43 102. Cost: 59
Edge: 42 90. Cost: 60
Edge: 41 95. Cost: 61
Edge: 42 84. Cost: 64
Edge: 81 102. Cost: 64
Edge: 115 116. Cost: 64
Edge: 11 102. Cost: 67
Edge: 30 47. Cost: 67

Edge: 27 56. Cost: 68
Edge: 93 111. Cost: 68
Edge: 5 87. Cost: 69
Edge: 6 111. Cost: 69
Edge: 39 121. Cost: 69
Edge: 45 60. Cost: 70
Edge: 60 137. Cost: 70
Edge: 24 36. Cost: 71
Edge: 15 91. Cost: 74
Edge: 29 32. Cost: 76
Edge: 19 38. Cost: 77
Edge: 50 118. Cost: 77
Edge: 71 86. Cost: 77
Edge: 101 127. Cost: 79
Edge: 6 18. Cost: 80
Edge: 27 64. Cost: 80
Edge: 73 84. Cost: 83
Edge: 25 58. Cost: 84
Edge: 106 107. Cost: 84
Edge: 80 91. Cost: 86
Edge: 125 126. Cost: 86
Edge: 67 70. Cost: 87
Edge: 71 80. Cost: 87
Edge: 3 51. Cost: 88
Edge: 1 20. Cost: 89
Edge: 46 80. Cost: 90
Edge: 128 129. Cost: 90
Edge: 54 80. Cost: 91
Edge: 11 113. Cost: 98
Edge: 20 79. Cost: 98
Edge: 9 28. Cost: 100
Edge: 40 78. Cost: 101
Edge: 4 73. Cost: 102
Edge: 33 39. Cost: 102
Edge: 9 63. Cost: 103
Edge: 93 114. Cost: 103
Edge: 41 49. Cost: 105
Edge: 21 88. Cost: 106
Edge: 19 90. Cost: 107
Edge: 25 82. Cost: 112
Edge: 43 71. Cost: 112
Edge: 2 41. Cost: 113
Edge: 61 119. Cost: 115
Edge: 61 103. Cost: 116
Edge: 100 104. Cost: 116
Edge: 11 60. Cost: 122
Edge: 33 56. Cost: 122
Edge: 51 99. Cost: 125
Edge: 61 138. Cost: 130
Edge: 119 120. Cost: 130
Edge: 4 88. Cost: 131

Edge: 63 101. Cost: 133
Edge: 63 119. Cost: 137
Edge: 100 101. Cost: 139
Edge: 10 55. Cost: 147
Edge: 85 99. Cost: 154
Edge: 14 22. Cost: 158
Edge: 48 123. Cost: 166
Edge: 5 43. Cost: 286
Total Cost: 9290

hsort:
C:\Users\clayw\source\repos\WarstlerC_Algorithms_Assignment4\HW4\code>mingw32-make hsort
C:/mingw64/bin/g++.exe -I./include/ -std=c++11 -DOPENCV=1 -IC:/opencv/build/install/include -std=c++11 src/algorithms/mst.cpp src/graph.cpp src/main.cpp src/sort/hsort.cpp -LC:/opencv/build/install/x64/mingw/bin -lopencv_imgcodecs3413 -lopencv_core3413 -lopencv_highgui3413 -lopencv_imgproc3413 -o bin/hsort
./bin/hsort
Perform unit test on the sorting algorithm
You are using Heap Sort algorithm
Your sorting implementation is correct

Perform unit test on your implementation with graph
Minimum Spanning Tree:
Edge: 1 0. Cost: 1
Edge: 1 2. Cost: 2
Edge: 3 1. Cost: 3
Edge: 2 4. Cost: 4
Edge: 5 4. Cost: 6
Total Cost: 16

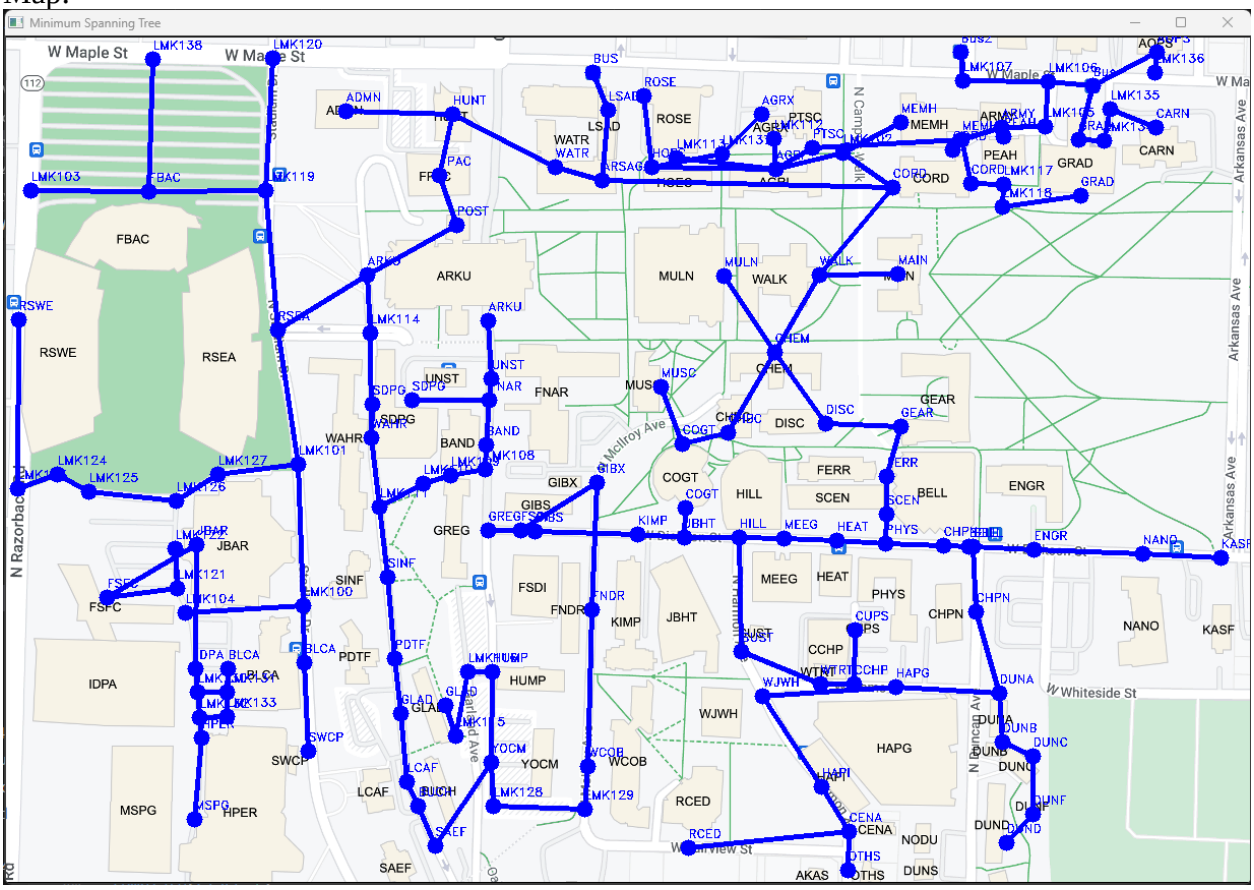
Minimum Spanning Tree:
Edge: 136 24. Cost: 20
Edge: 72 29. Cost: 21
Edge: 118 117. Cost: 22
Edge: 56 130. Cost: 23
Edge: 131 74. Cost: 23
Edge: 75 108. Cost: 24
Edge: 1 116. Cost: 24
Edge: 134 17. Cost: 25
Edge: 47 65. Cost: 26
Edge: 133 132. Cost: 27
Edge: 98 107. Cost: 28
Edge: 109 110. Cost: 28
Edge: 131 130. Cost: 29
Edge: 92 44. Cost: 29
Edge: 57 96. Cost: 29
Edge: 42 16. Cost: 29
Edge: 11 112. Cost: 30
Edge: 134 135. Cost: 31

Edge: 115 89. Cost: 31
Edge: 117 94. Cost: 32
Edge: 3 23. Cost: 32
Edge: 34 93. Cost: 33
Edge: 58 0. Cost: 33
Edge: 8 97. Cost: 33
Edge: 108 109. Cost: 34
Edge: 124 125. Cost: 35
Edge: 92 83. Cost: 37
Edge: 22 37. Cost: 38
Edge: 69 76. Cost: 38
Edge: 87 26. Cost: 39
Edge: 121 122. Cost: 39
Edge: 7 10. Cost: 40
Edge: 55 11. Cost: 41
Edge: 124 123. Cost: 41
Edge: 65 79. Cost: 42
Edge: 129 85. Cost: 42
Edge: 10 94. Cost: 43
Edge: 105 7. Cost: 43
Edge: 128 20. Cost: 43
Edge: 105 106. Cost: 44
Edge: 106 36. Cost: 44
Edge: 29 75. Cost: 44
Edge: 52 82. Cost: 44
Edge: 40 96. Cost: 46
Edge: 54 35. Cost: 46
Edge: 2 5. Cost: 47
Edge: 133 74. Cost: 47
Edge: 111 110. Cost: 48
Edge: 62 44. Cost: 48
Edge: 59 135. Cost: 48
Edge: 126 127. Cost: 48
Edge: 97 73. Cost: 48
Edge: 15 83. Cost: 50
Edge: 95 28. Cost: 51
Edge: 22 21. Cost: 51
Edge: 62 52. Cost: 52
Edge: 0 77. Cost: 53
Edge: 36 17. Cost: 54
Edge: 96 82. Cost: 54
Edge: 18 30. Cost: 54
Edge: 137 12. Cost: 55
Edge: 67 100. Cost: 56
Edge: 16 44. Cost: 57
Edge: 9 114. Cost: 57
Edge: 66 72. Cost: 57
Edge: 69 8. Cost: 57
Edge: 102 43. Cost: 59
Edge: 35 53. Cost: 59
Edge: 90 42. Cost: 60

Edge: 95 41. Cost: 61
Edge: 115 116. Cost: 64
Edge: 102 81. Cost: 64
Edge: 84 42. Cost: 64
Edge: 11 102. Cost: 67
Edge: 47 30. Cost: 67
Edge: 111 93. Cost: 68
Edge: 27 56. Cost: 68
Edge: 5 87. Cost: 69
Edge: 121 39. Cost: 69
Edge: 111 6. Cost: 69
Edge: 60 45. Cost: 70
Edge: 137 60. Cost: 70
Edge: 36 24. Cost: 71
Edge: 15 91. Cost: 74
Edge: 32 29. Cost: 76
Edge: 71 86. Cost: 77
Edge: 118 50. Cost: 77
Edge: 19 38. Cost: 77
Edge: 127 101. Cost: 79
Edge: 64 27. Cost: 80
Edge: 6 18. Cost: 80
Edge: 84 73. Cost: 83
Edge: 58 25. Cost: 84
Edge: 107 106. Cost: 84
Edge: 126 125. Cost: 86
Edge: 91 80. Cost: 86
Edge: 71 80. Cost: 87
Edge: 70 67. Cost: 87
Edge: 51 3. Cost: 88
Edge: 1 20. Cost: 89
Edge: 128 129. Cost: 90
Edge: 46 80. Cost: 90
Edge: 54 80. Cost: 91
Edge: 113 11. Cost: 98
Edge: 20 79. Cost: 98
Edge: 28 9. Cost: 100
Edge: 78 40. Cost: 101
Edge: 33 39. Cost: 102
Edge: 4 73. Cost: 102
Edge: 9 63. Cost: 103
Edge: 114 93. Cost: 103
Edge: 49 41. Cost: 105
Edge: 21 88. Cost: 106
Edge: 90 19. Cost: 107
Edge: 71 43. Cost: 112
Edge: 25 82. Cost: 112
Edge: 41 2. Cost: 113
Edge: 61 119. Cost: 115
Edge: 61 103. Cost: 116
Edge: 104 100. Cost: 116

Edge: 33 56. Cost: 122
Edge: 60 11. Cost: 122
Edge: 51 99. Cost: 125
Edge: 61 138. Cost: 130
Edge: 120 119. Cost: 130
Edge: 4 88. Cost: 131
Edge: 63 101. Cost: 133
Edge: 63 119. Cost: 137
Edge: 101 100. Cost: 139
Edge: 55 10. Cost: 147
Edge: 99 85. Cost: 154
Edge: 14 37. Cost: 158
Edge: 123 48. Cost: 166
Edge: 5 43. Cost: 286
Total Cost: 9290

Map:



Conclusions:

Overall, everything worked as expected. The program creates a map of the campus and then outlines the shortest path