Assignment 2

Kelly Jo Law and Riley Primeau

October 30, 2022

1 Task 1

See file:

2 Task 2

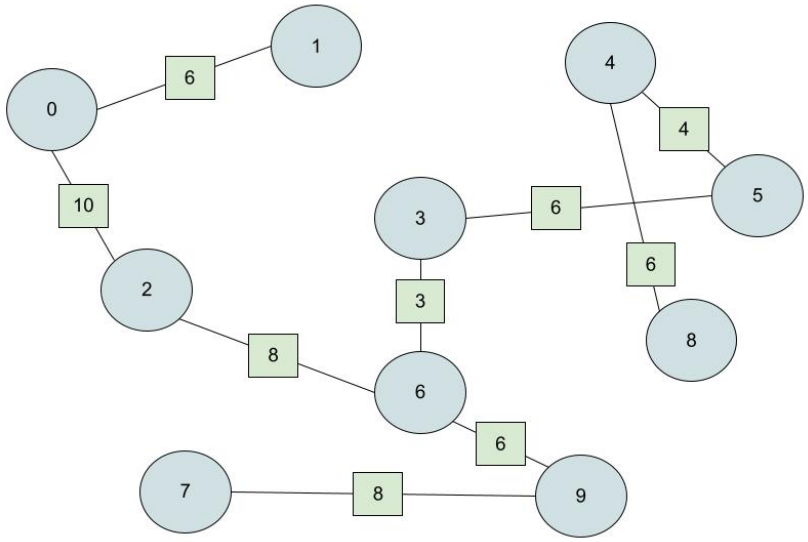
2.1 Express this graph as an adjacency matrix.

*	0	1	2	3	4	5	6	7	8	9
0	0	6	10	0	0	0	0	0	0	0
1	6	0	12	11	14	0	0	0	0	0
2	10	12	0	12	0	0	8	16	0	0
3	0	11	12	0	0	6	3	0	0	0
4	0	14	0	0	0	4	0	0	6	0
5	0	0	0	6	4	0	0	0	12	0
6	0	0	8	3	0	0	0	0	16	6
7	0	0	16	0	0	0	0	0	0	8
8	0	0	0	0	6	12	16	0	0	13
9	0	0	0	0	0	0	6	8	13	0

2.2 Use your adjacency matrix as input to a C++ program (Blacklisted - STL) that derives the minimum spanning tree by applying Prim's algorithm (Levitin), your output should in the form of an edge list (one triplet per line). Once concluded the program should report time taken and the number of comparisons.

See file: primsalgorithm.cpp

2.3 Use your edge list to manually colorize (try paint or photoshop) the graph given indicating the path for visual inspection/verification.



3 Task 3

Design an efficient algorithm for finding and deleting an element of the smallest value in a heap and determine its time efficiency (clear and correctly styled PseudoCode).

Finds the minimum node in the heap, deletes the node, and sifts to restore the heap
Input: heap
Output: heap with the smallest node removed

Begin pseudocode:

Function to find the minimum node in the heap

```
findMinNode()
// variable declarations
begin IF
    IF rightIndex ≥ heapSize
        begin IF
        IF leftIndex ≥ heapSize
            return;
        ELSE minIndex = leftIndex;
        end IF
    ELSE
    begin IF
        IF data[leftIndex] ≤ data[rightIndex]
            minIndex = leftIndex;
        ELSE minIndex = rightIndex;
    end IF
begin IF
    IF data[node] > data[minIndex]
        temp = data[minIndex];
    end IF
end function
```

Function to remove minimum node

```
removeMIn()
    begin IF
    IF (isEmpty)
```

```
        throw "empty"
    ELSE
        data[0] = heapSize - 1;
        - - heapSize
        begin IF
        IF heapsize > 0
            siftdown(0);
end function
```