 **University of the Western Cape**

**Department of Computer Science**

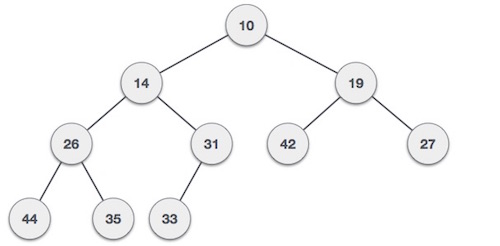
**Private Bag X17, Bellville, 7535**

**CSC211- Data Structures and Algorithms II**

**Assignment 3 (25 Marks) Student number: 3 8 5 1 2 7 2**

-----------------------------------------------------------------------------------------------------------------------------------------

1. Consider the min heap below. Perform 5 deletion operations and show the resulting heap after each deletion. **[5]**



**Result after 1st deletion of min node(10)**

**-----🡪**

A close up of text on a whiteboard

Description automatically generated

19

**Result after 2nd deletion of min node(14)**

**-----🡪**

14

27

26

19

33

33

26

31

42

27

31

42

35

35

44

44

**Result after 3rd deletion of min node(26)**

**-----🡪**

26

**Result after 3rd deletion of min node(19)**

**-----🡪**

27

27

31

31

35

42

35

33

42

33

44

35

42

31

**Result after 3rd deletion of min node(27)**

**-----🡪**

44

33

44

1. Sort the following list from **largest** **to** **smallest** using *heapsort*. Represent the heap by an array and show each step: S, O, R, T, I, N, G (A > B, B > C, C > D…. Y>Z). **[7**]

G

S

N

I

T

R

O

A close up of text on a white background

Description automatically generated

T

R

S

I

N

G

G

N

I

T

R

O

S

S

O

R

T

I

N

G

G

N

I

T

R

O

S

T

O

R

S

I

N

G

O

G

N

I

S

R

O

T

G

N

I

S

R

O

T

1. Consider the graph below, compute the **Positive-Weighted Shortest Path** from **A to G**. Please show your adjacency matrix**. [5]**

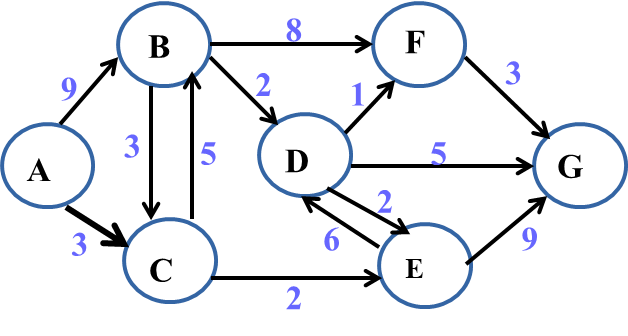
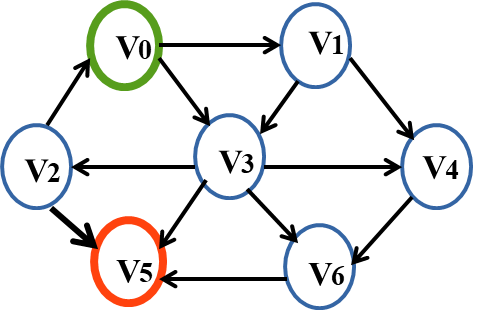


Figure 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F | G |
| A | ∞ | 9 | 3 | 0 | 0 | 0 | 0 |
| B | 0 | ∞ | 3 | 2 | 0 | 8 | 0 |
| C | 0 | 5 | ∞ | 0 | 2 | 0 | 0 |
| D | 0 | 0 | 0 | ∞ | 2 | 1 | 5 |
| E | 0 | 0 | 0 | 6 | ∞ | 0 | 9 |
| F | 3 | 0 | 0 | 0 | 0 | ∞ | 3 |
| G | 0 | 0 | 0 | 0 | 0 | 0 | ∞ |

**The shortest positive weighted path:** A + C + E + G = 14.

1. For the graph below, find the **unweighted shortest path** from V0 to V5, using Breadth-First Search. Please also show your Breadth First Tree. **[5]**



Therefore, with reference to the tree we can see that the shortest arc length is 2.

V0

V­1

V­1

V1

V3

V3

V3

V­4

V­3

V3

V2

V6

V5

V­6

V5

V6

V5

V5

V5

V­5

1. In **at most** **3 sentences**, differentiate between breadth-first traversal and depth-first traversal of graphs. [3]

DFS is implemented by a stack while BFS is implemented by a queue. DFS traverses as far deep into the graph, then backs up, BFS traverses to the closest vertice, then backs up to the next vertice with equal path length before carrying on to deepest vertice. Finally DFS follows the last-in-first-out policy whereas BFS follows the first-in-first-out policy.