ĐẠI HỌC TÔN ĐỨC THẮNG TON DUC THANG UNIVERSITY	FACULTY OF INFORMATION TECHNOLOGY FINAL EXAMINATION SEMESTER 1 ACADEMIC YEAR: 2023 - 2024

SUBJECT: DATA STRUCTURES AND ALGORITHMS
SUBJECT CODE: 504008

INSTRUCTION TO PRESENT ALGORITHMS IN THE FINAL EXAMINATION

I. SORTING ALGORITHMS

Selection Sort, Bubble Sort, and Insertion Sort are presented as below
Given a = {6, 9, 5, 8, 1, 3},

Insertion Sort

pass	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]
-	6	9	5	8	1	3
1	6	9	5	8	1	3
2	5	6	9	8	1	3
3	5	6	8	9	1	3
4	1	5	6	8	9	3
5	1	3	5	6	8	9

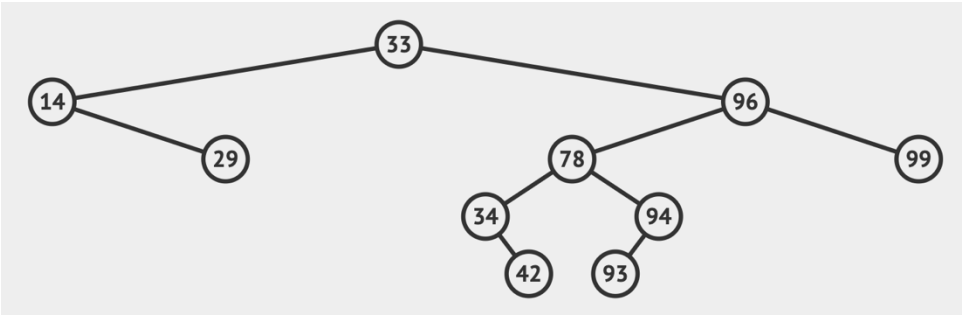
Note

- Pass "-" presents the original array
- A pass is one iteration step of the outer loop

II. BINARY SEARCH TREE & AVL TREE

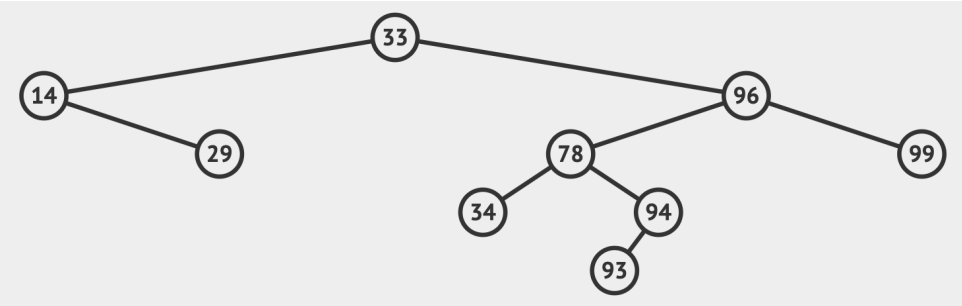
a) Binary Search Tree

Students draw the final tree after inserting all given keys in.

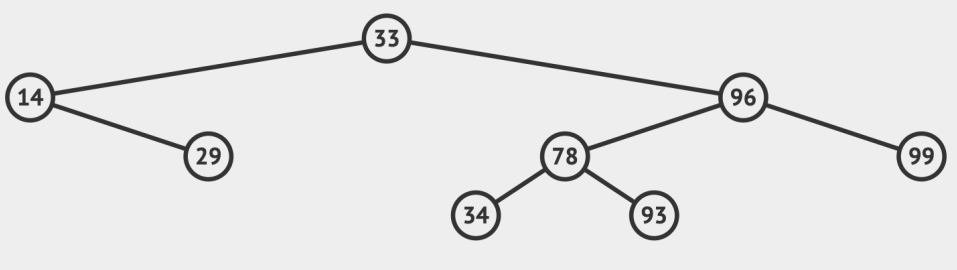


For every operation that modifies the tree structure, students draw a new figure to illustrate the result. For example,

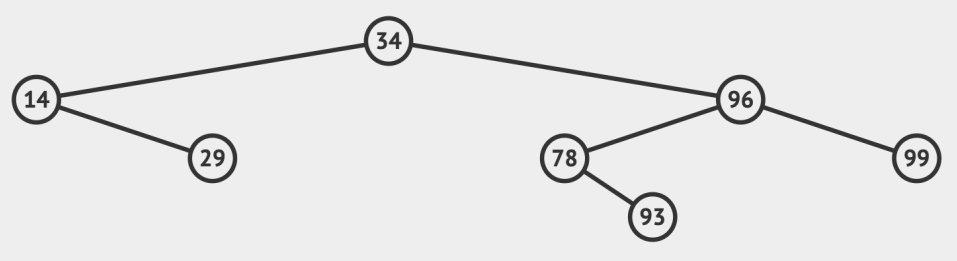
- "*delete node 42*"



- "*delete node 94*"



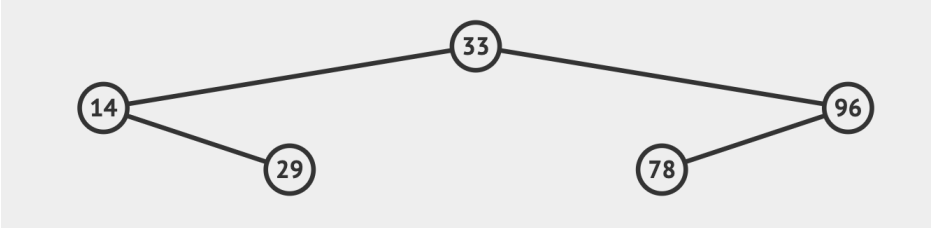
- "*delete node 33 using successors*"



b) AVL Tree

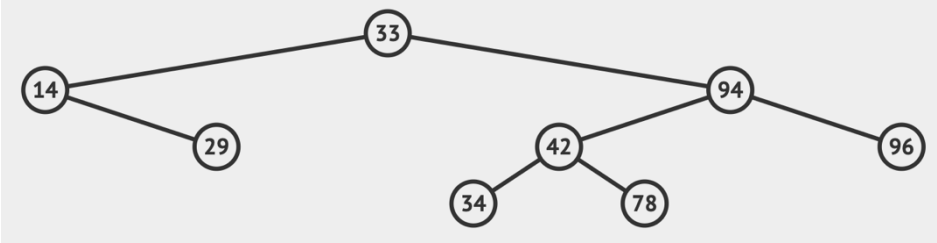
Students iteratively add given keys to the AVL tree then draw a figure after adding a key that causes rotation. For example, "*create an AVL tree given a list of keys [33, 96, 14, 29, 78, 94, 34, 42, 93, 99]*"

- *Add 33, 96, 14, 29, 78*

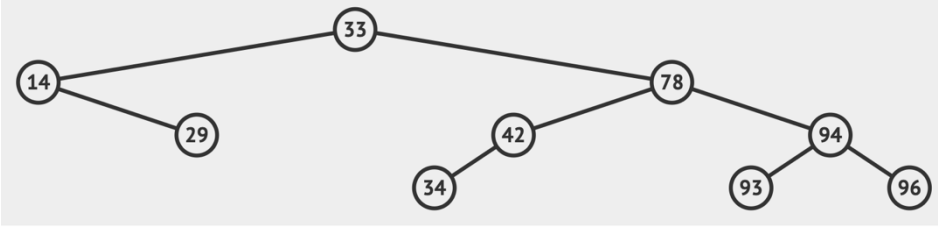


- *Add 94*

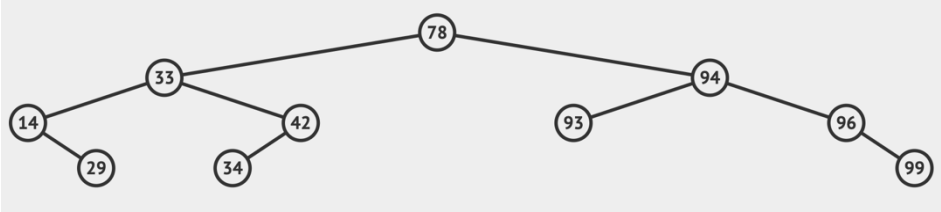
- *Add 34, 42*



- *Add 93*



- *Add 99*



III. HEAP

Given a list of keys

33 96 14 29 78 94 34 42 93 99

Present the procedure to build a Binary Min Heap where iteratively adding key by key into the heap.

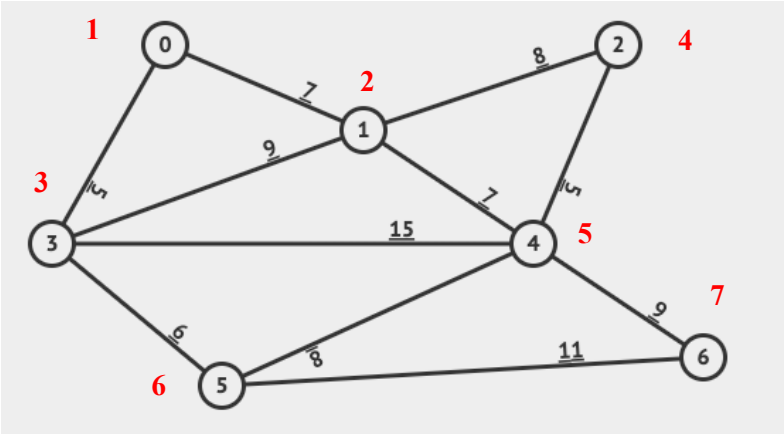
Step	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
1.	33									
2.	33	96								
3.	14	96	33							
4.	14	29	33	96						
5.	14	29	33	96	78					
6.	14	29	33	96	78	94				

7.	14	29	33	96	78	94	34			
8.	14	29	33	42	78	94	34	96		
9.	14	29	33	42	78	94	34	96	93	
10.	14	29	33	42	78	94	34	96	93	99

IV. GRAPH TRAVERSAL

Students run BFS, DFS algorithms and then write down the list of vertices in the traversal order. For example,

BFS: 0-1-3-2-4-5-6

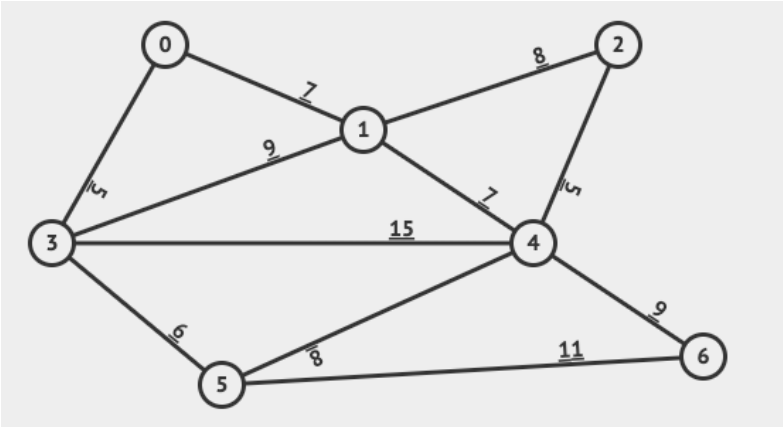


Graph traversal using BFS algorithm

V. MINIMUM SPANNING TREE

a) Kruskal's algorithm

Given a graph below, students draw a table as below to visualize Kruskal's algorithm.

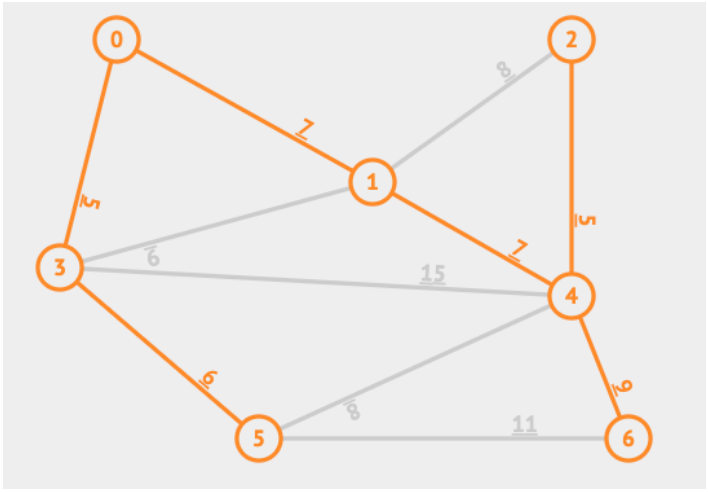


Obviously, edges are sorted in the ascending of weights.

Edge	Selected?
5, 0-3	Yes
5, 2-4	Yes
6, 3-5	Yes

7, 0-1	Yes
7, 1-4	Yes
8, 1-2	No
8, 4-5	No
9, 1-3	No
9, 4-6	Yes
11, 5-6	No
15, 3-4	No

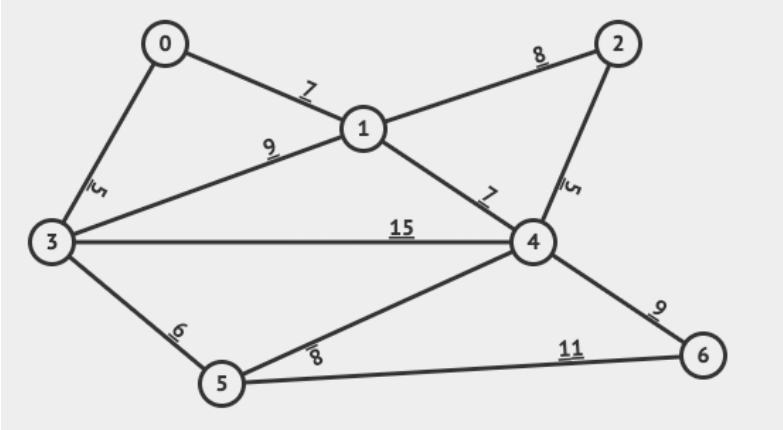
Students draw the minimum spanning tree (selected edges only).



Total cost: 39.

a) Prim's algorithm

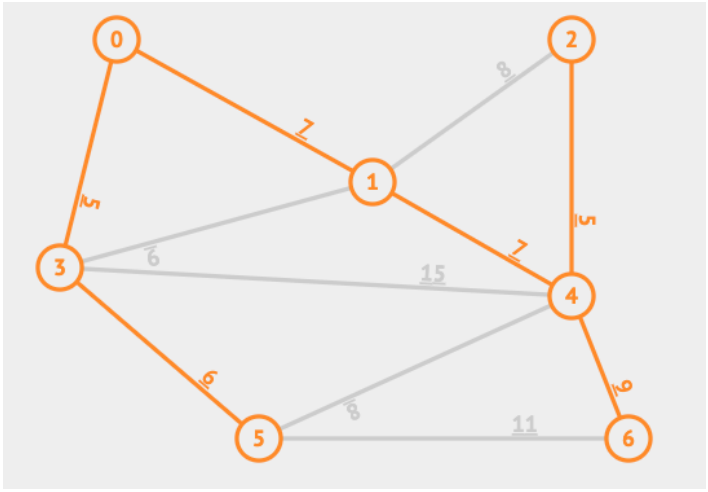
Given a graph below, students draw a table as below to present Prim's algorithm.



Dequeued item	*Selected?*
5, 0-3	*Yes*
6, 3-5	*Yes*
7, 0-1	*Yes*

7, 1-4	*Yes*
5, 4-2	*Yes*
8, 1-2	*No*
8, 5-4	*No*
9, 3-1	*No*
9, 4-6	*Yes*
11, 5-6	*No*
15, 3-4	*No*

Students draw the minimum spanning tree (selected edges only).

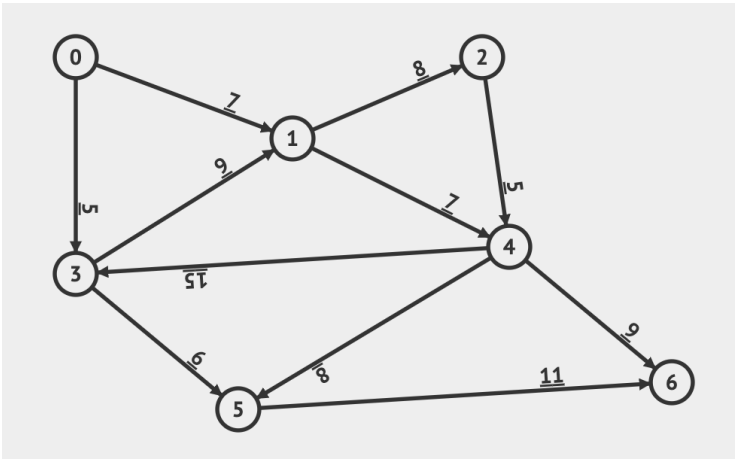


Total cost: 39.

VI. SINGLE-SOURCE SHORTEST PATHS

a) Bellman-Ford's algorithm

Given a graph below, students draw a table to present Bellman Ford's algorithm, starting from vertex 0.



	0	1	2	3	4	5	6
-	-1, 0	-1, ∞	-1, ∞	-1, ∞	-1, ∞	-1, ∞	-1, ∞
Pass 1	-1, 0	0, 7	1, 15	0, 5	1, 14	3, 11	5, 22
Pass 2	-1, 0	0, 7	1, 15	0, 5	1, 14	3, 11	5, 22
Pass 3	-1, 0	0, 7	1, 15	0, 5	1, 14	3, 11	5, 22
Pass 4	-1, 0	0, 7	1, 15	0, 5	1, 14	3, 11	5, 22
Pass 5	-1, 0	0, 7	1, 15	0, 5	1, 14	3, 11	5, 22
Pass 6	-1, 0	0, 7	1, 15	0, 5	1, 14	3, 11	5, 22

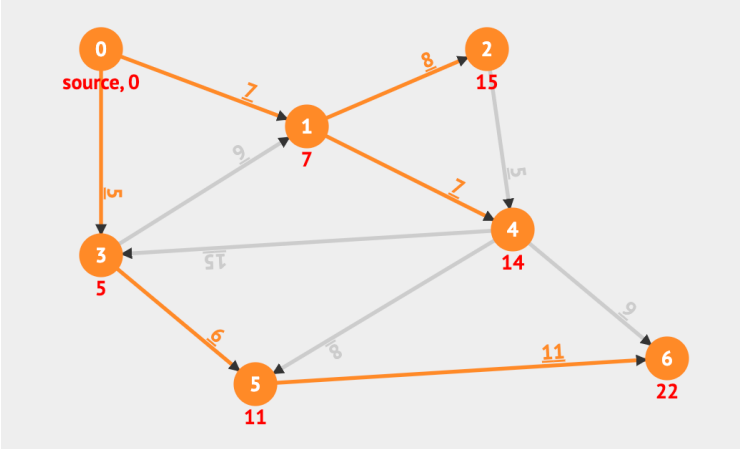
Each row in table is a pass of the algorithm. There are no changes from pass 2, so students can stop the algorithm after pass 2 and skip drawing the remained rows.

Finally, students write all shortest paths down using the following ways for ease.

0 – 1 – 2

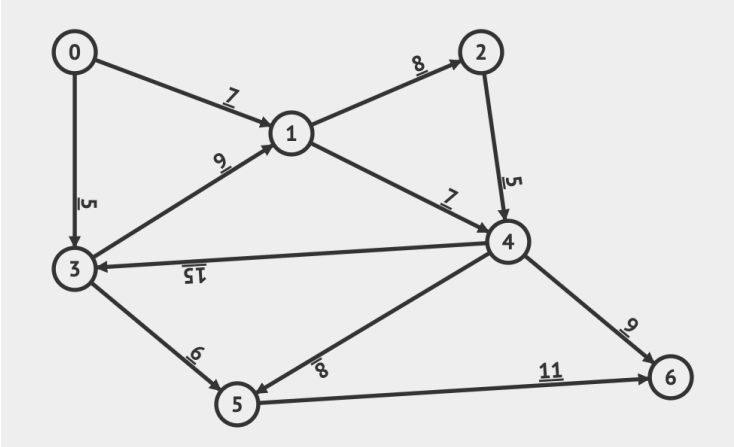
0 – 1 – 4

0 – 3 – 5 – 6



b) Dijkstra's algorithm

Given a graph below, students draw a table to present Dijkstra's algorithm, starting from vertex 0.



	0	1	2	3	4	5	6
0:	-1, 0	-1, ∞	-1, ∞	-1, ∞	-1, ∞	-1, ∞	-1, ∞
3:	-1, 0	0, 7	-1, ∞	0, 5	-1, ∞	-1, ∞	-1, ∞
1:	-1, 0	0, 7	-1, ∞	0, 5	-1, ∞	3, 11	-1, ∞
5:	-1, 0	0, 7	1, 15	0, 5	1, 14	3, 11	-1, ∞
4:	-1, 0	0, 7	1, 15	0, 5	1, 14	3, 11	5, 22
2:	-1, 0	0, 7	1, 15	0, 5	1, 14	3, 11	5, 22
6:	-1, 0	0, 7	1, 15	0, 5	1, 14	3, 11	5, 22

Each row in the table is an iteration step of the algorithm (dequeue from the Priority Queue). Students write in cells that are updated and leave the remained cells blank for each step. There must be enough rows that corresponding to iteration steps despite there are several blank ones.

Finally, students write all shortest paths down using the following ways for ease.

0 – 1 – 2

0 – 1 – 4

0 – 3 – 5 – 6