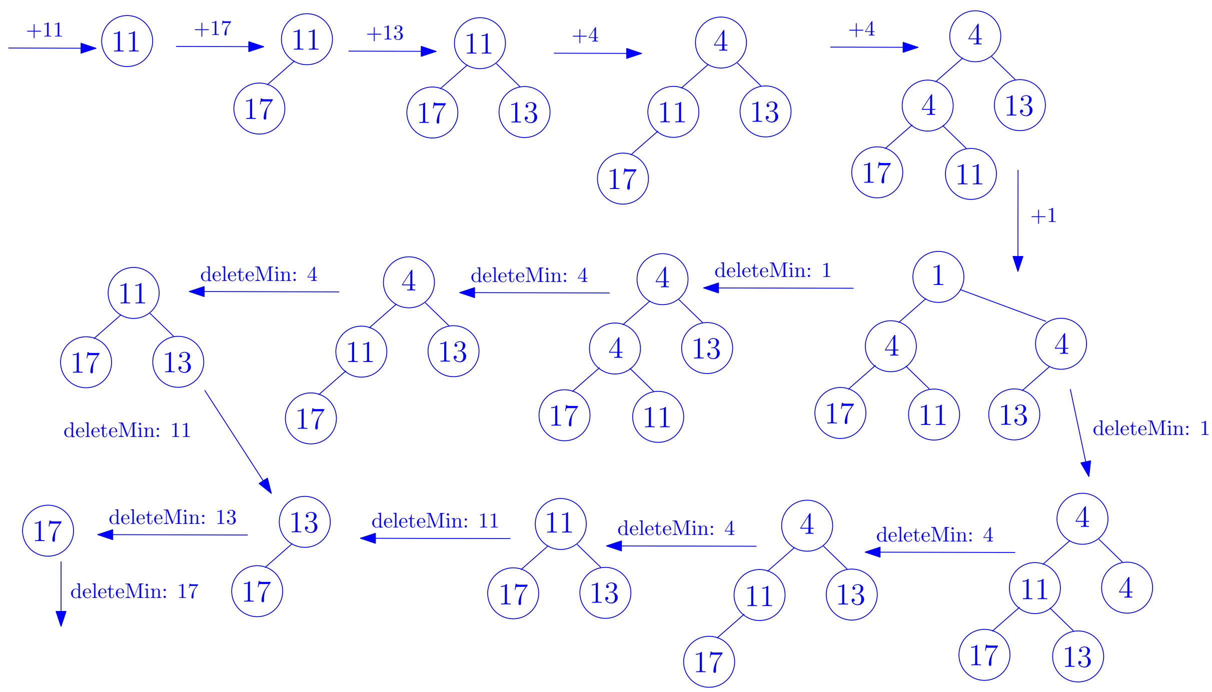
Practice Set 9 (Heap and Dijkstra's Algorithm)

Data Structure (CS 223)

1 Heap

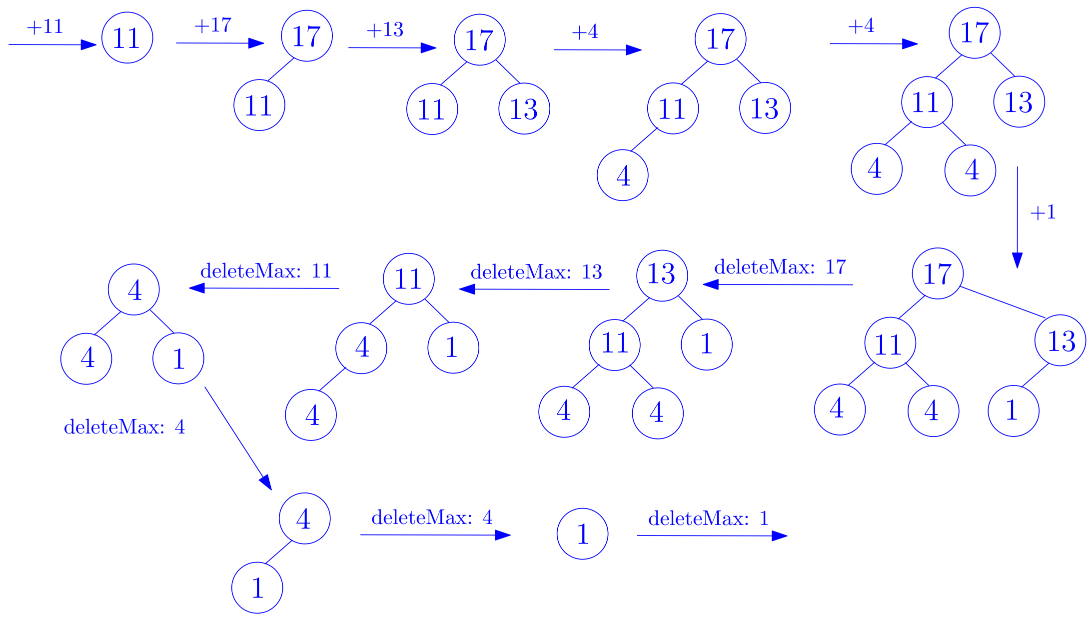
Q1: Show the heap **at each stage** when the following numbers are **inserted** to an initially empty min-heap in the given order: {*11, 17, 13, 4, 4, 1*}. Now, show the heap **at each stage** when we successively perform the **deleteMin** operation on the heap until it is empty.

Answer:



Q2: In a max-heap, the heap property changes to – *the value at a node is greater than or equal to the value of its children*. Deletion and insertion have to maintain this changed heap property. Answer Q1 when the min-heap is replaced by a max-heap, and **deleteMin** by **deleteMax**.

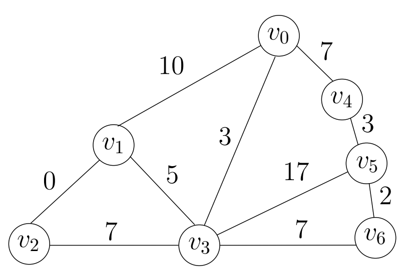
Answer: See next page.



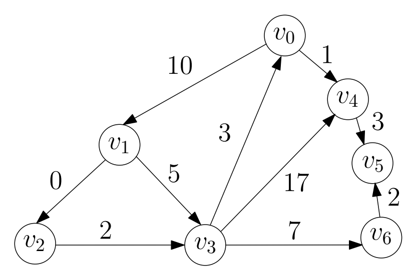
2 Dijkstra's Algorithm

Q3: Starting from node v_1, illustrate Dijkstra's algorithm for finding the shortest paths in Graph A and in Graph B. Show at each stage:

- the distance and parent arrays,
- the open and closed sets, and
- the vertex chosen for relaxation.



Graph A



Graph B

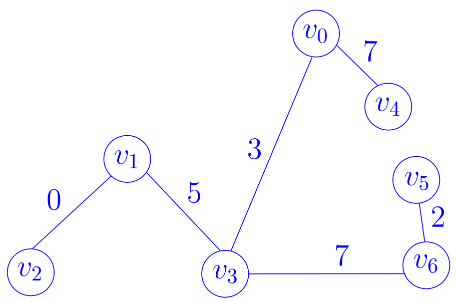
Use the *parent* array above to draw the shortest path tree in each case.

Undirected Graph

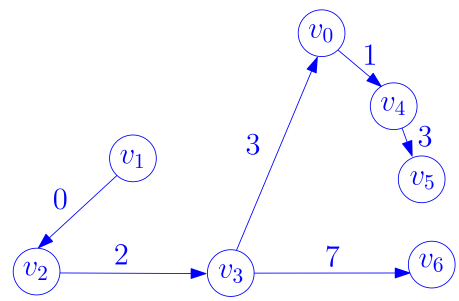
	dist							open	closed	parent						
	v_0	v_1	v_2	v_3	v_4	v_5	v_6			v_0	v_1	v_2	v_3	v_4	v_5	v_6
At Start:	∞	0	∞	∞	∞	∞	∞	$\{v_1\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
Relax v_1:	10	0	0	5	∞	∞	∞	$\{v_2, v_3, v_0\}$	$\{v_1\}$	v_1	$\varnothing$	v_1	v_1	$\varnothing$	$\varnothing$	$\varnothing$
Relax v_2:	10	0	0	5	∞	∞	∞	$\{v_3, v_0\}$	$\{v_1, v_2\}$	v_1	$\varnothing$	v_1	v_1	$\varnothing$	$\varnothing$	$\varnothing$
Relax v_3:	8	0	0	5	∞	22	12	$\{v_0, v_6, v_5\}$	$\{v_1, v_2, v_3\}$	v_3	$\varnothing$	v_1	v_1	$\varnothing$	v_3	v_3
Relax v_0	8	0	0	5	15	22	12	$\{v_6, v_4, v_5\}$	$\{v_1, v_2, v_3, v_0\}$	v_3	$\varnothing$	v_1	v_1	v_0	v_3	v_3
Relax v_6	8	0	0	5	15	14	12	$\{v_5, v_4\}$	$\{v_1, v_2, v_3, v_0, v_6\}$	v_3	$\varnothing$	v_1	v_1	v_0	v_6	v_3
Relax v_5	8	0	0	5	15	14	12	$\{v_4\}$	$\{v_1, v_2, v_3, v_0, v_6, v_5\}$	v_3	$\varnothing$	v_1	v_1	v_0	v_6	v_3
Relax v_4	8	0	0	5	15	14	12	$\varnothing$	$\{v_1, v_2, v_3, v_0, v_6, v_5, v_4\}$	v_3	$\varnothing$	v_1	v_1	v_0	v_6	v_3

Directed Graph

	dist							open	closed	parent						
	v_0	v_1	v_2	v_3	v_4	v_5	v_6			v_0	v_1	v_2	v_3	v_4	v_5	v_6
At Start:	∞	0	∞	∞	∞	∞	∞	$\{v_1\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
Relax v_1:	∞	0	0	5	∞	∞	∞	$\{v_2, v_3\}$	$\{v_1\}$	$\varnothing$	$\varnothing$	v_1	v_1	$\varnothing$	$\varnothing$	$\varnothing$
Relax v_2:	∞	0	0	2	∞	∞	∞	$\{v_3\}$	$\{v_1, v_2\}$	$\varnothing$	$\varnothing$	v_1	v_2	$\varnothing$	$\varnothing$	$\varnothing$
Relax v_3:	5	0	0	2	19	∞	9	$\{v_0, v_6, v_4\}$	$\{v_1, v_2, v_3\}$	v_3	$\varnothing$	v_1	v_2	v_3	$\varnothing$	v_3
Relax v_0	5	0	0	2	6	∞	9	$\{v_4, v_6\}$	$\{v_1, v_2, v_3, v_0\}$	v_3	$\varnothing$	v_1	v_2	v_0	$\varnothing$	v_3
Relax v_4	5	0	0	2	6	9	9	$\{v_6, v_5\}$	$\{v_1, v_2, v_3, v_0, v_4\}$	v_3	$\varnothing$	v_1	v_2	v_0	v_4	v_3
Relax v_6	5	0	0	2	6	9	9	$\{v_5\}$	$\{v_1, v_2, v_3, v_0, v_4, v_6\}$	v_3	$\varnothing$	v_1	v_2	v_0	v_4	v_3
Relax v_5	5	0	0	2	6	9	9	$\varnothing$	$\{v_1, v_2, v_3, v_0, v_4, v_6, v_5\}$	v_3	$\varnothing$	v_1	v_2	v_0	v_4	v_3



Shortest Path Tree for Graph A



Shortest Path Tree for Graph B

Q4: Answer the following questions for Dijkstra's algorithm executed on an undirected graph with N vertices and M edges:

- What is the maximum number of times an edge is looked at?

  Answer: Twice; once each when the vertex at either end is removed from open.

- What is the maximum number of times the distance value of a vertex is updated?

  Answer: As many times as the number of incoming edges.

- At any point, what is the maximum number of open vertices? Draw a graph on 4 vertices for which this maximum number is realized, irrespective of the starting node.

  Answer: $N - 1$. A graph with 4 vertices, where any two vertices have an edge between them.

Q5: Provide a good intuition/reasoning as to why when a vertex v is closed, we will never find a shorter path to v thereafter.

Answer: All edge weights are non-negative, and Dijkstra's algorithm relaxes (closes) vertices in increasing order of shortest path distance from s. Hence, once a vertex v is closed, if we extend the shortest path from s to v, its weight can never decrease.