Homework 6 (COMPSCI 223)

Q1: Answer the following:

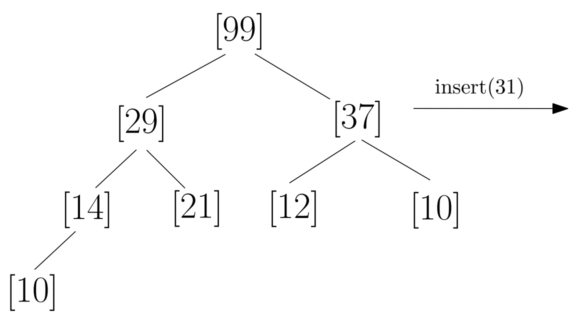
- Starting with the heap array **[10, 15, 28, 16]**, show the final heap array after we insert 1

  Answer: [1, 10, 28, 16, 15]

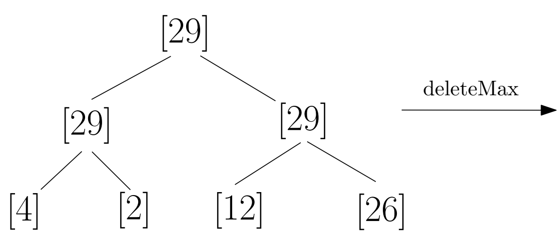
- Starting with the heap array **[-1, 2, 7, 6, 18, 92]**, show the final heap array after we delete the minimum.

  Answer: [2, 6, 7, 92, 18]

Q2: For each of the following MAX heaps, show the final heap array when the adjoining operation is performed.



Answer: [99, 31, 37, 29, 21, 12, 10, 10, 14]



Answer: [29, 26, 29, 4, 2, 12] or [29, 29, 26, 4, 2, 12]

Q3: Write true or false for the following:

- In the worst-case, the complexity of inserting in a heap having N nodes is $O(\log N)$.

  True.

- For sorting N numbers, the worst case complexity of heap sort is $O(N^2)$, whereas the best case complexity is $O(N \log N)$.

  False. Both $O(N \log N)$

- Given a min-heap containing only distinct numbers, the second minimum number in a min-heap is a child of the root node.

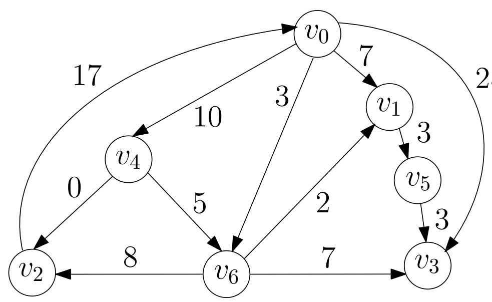
  True.

- Suppose you have a max-heap, where all numbers are distinct. Then, the third highest number is definitely a child of the root node.

  False. Depends on the structure. The third highest can be a child of the second highest or a child of the maximum.

Q4: The best-case complexity of inserting a number into a max-heap is $O(1)$; for example, when you insert a new minimum.

Q5: Starting from node v_0, illustrate Dijkstra's algorithm on the following graph. At each stage, show **the relaxed vertex**, **the open and closed** sets, and the **dist and parent** arrays.



	open	**closed**	**dist array**							**parent array**						
			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
Start:	v_0		0	∞	∞	∞	∞	∞	∞	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
relax v_0	v_1,v_3, v_4, v_6	v_0	0	7	∞	25	10	∞	3	$\varnothing$	v_0	$\varnothing$	v_0	v_0	$\varnothing$	v_0
relax v_6	v_1, v_2, v_3, v_4	v_0, v_6	0	5	11	10	10	∞	3	$\varnothing$	v_6	v_6	v_6	v_0	$\varnothing$	v_0
relax v_1	v_2, v_3, v_4, v_5	v_0,v_6,v_1	0	5	11	10	10	8	3	$\varnothing$	v_6	v_6	v_6	v_0	v_1	v_0
relax v_5	v_2, v_3, v_4	v_0, v_6, v_1, v_5	0	5	11	10	10	8	3	$\varnothing$	v_6	v_6	v_6	v_0	v_1	v_0
relax v_3	v_2, v_4	v_0, v_6, v_1, v_5, v_3	0	5	11	10	10	8	3	$\varnothing$	v_6	v_6	v_6	v_0	v_1	v_0
relax v_4	v_2	v_0, v_6, v_1, v_5, v_3, v_4	0	5	10	10	10	8	3	$\varnothing$	v_6	v_4	v_6	v_0	v_1	v_0
relax v_2		v_0, v_6, v_1, v_5, v_3, v_4, v_2	0	5	10	10	10	8	3	$\varnothing$	v_6	v_4	v_6	v_0	v_1	v_0

Q6: How many times is a vertex relaxed in Dijkstra's algorithm?

- At most once $\longleftarrow$ **Answer**
- As many times as the number of vertices in the graph
- As many times as the number of edges in the graph
- Depends on the professor's mood

Q7: Dijkstra's algorithm finds (choose the most appropriate):

- a shortest path from a source to a destination
- shortest paths from a source to every vertex $\longleftarrow$ **Answer**
- shortest paths from a source to all vertices other than source
- shortest paths from every vertex to every vertex

Q8: Given a directed weighted graph, *is there a path from a vertex s to a vertex t whose weight is at most W*? We can use Dijkstra's algorithm to answer this question as follows:

- Run Dijkstra from t. The answer is YES if $dist[s] > W$, else NO.
- Run Dijkstra from t. The answer is NO if $dist[s] \leq W$, else YES.
- Run Dijkstra from s. The answer is NO if $dist[s] > W$, else YES.
- Run Dijkstra from s. The answer is NO if $dist[t] > W$, else YES. $\longleftarrow$ **Answer**

Q9: A vertex is closed more than once in Dijkstra's algorithm. True or False?

Answer: False

Q10: You are given a pre-built max-heap having N numbers. Suppose the maximum number in the heap occurs k times. Give an $O(k \log N)$ time algorithm to compute k. **Write pseudo-code.**

Answer: Just keep deleting the maximum until you get a new number. More specifically,

- $max = deleteMax()$
- $k = 1$
- while $(size() \neq 0$ and max equals $deleteMax())$
  - $k++$
- return k