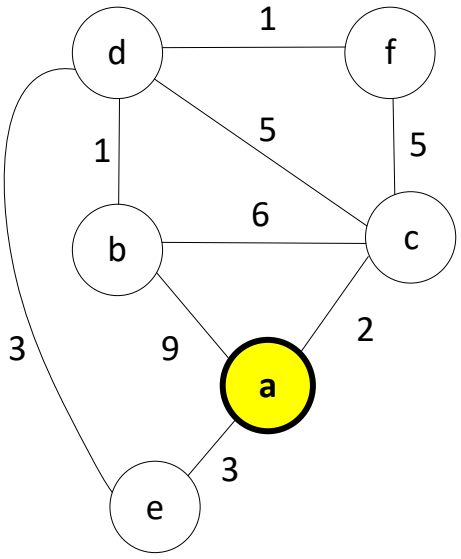
1. Dijkstra's Algorithm (12 pts.)

Suppose Dijkstra's algorithm is run on the graph below using **a** as the starting node. For the first 4 iterations of the primary loop in Dijkstra's algorithm (which you should know or understand functionally), show which node will be returned by a call to top() on the heap AND the heap values (priorities of each node) just BEFORE pop() is called.



We have shown the heap just before the 1st call to top()/pop() in the first iteration of the primary loop. Complete the contents of the table below for the next 3 iterations.

Iteration	1	2	3	4
top() node	**a**	**c**	**e**	**d**

Heap contents (in no particular order before pop())
 	
Note:
`inf` = infinity
	`a : 0`
	`b : inf`
	`c : inf`
	`d : inf`
	`e : inf`
	`f : inf`
		`b: 9`
		`c: 2`
		`d: infinity`
		`e: 3`
		`f: infinity`
		
		`Removed:`
		`a: 0 (not required)`
			`b: 8`
			`d: 7`
			`e: 3`
			`f: 7`
			
			`Removed:`
			`a: 0 (not required)`
			`c: 2 (not required)`
				`b: 8`
				`d: 6`
				`f: 7`
				
				`Removed:`
				`a: 0 (not required)`
				`c: 2 (not required)`
				`e: 3 (not required)`