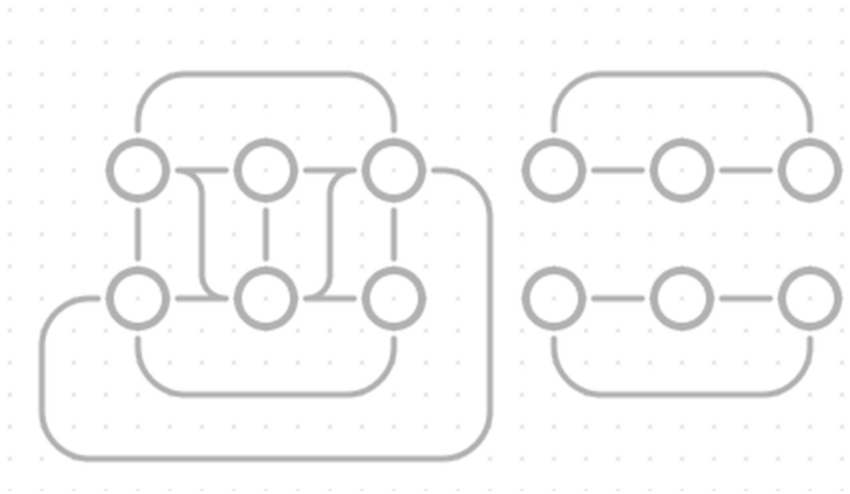
CSCI 230 -- Homework 3

Nero Li

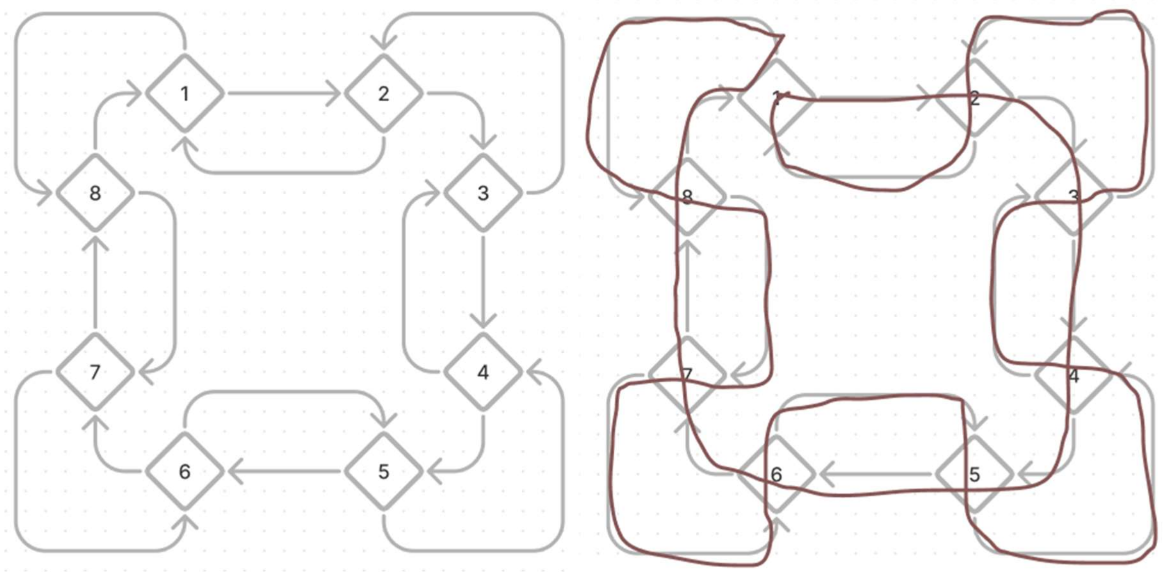
Chapter 13

- R-13.1

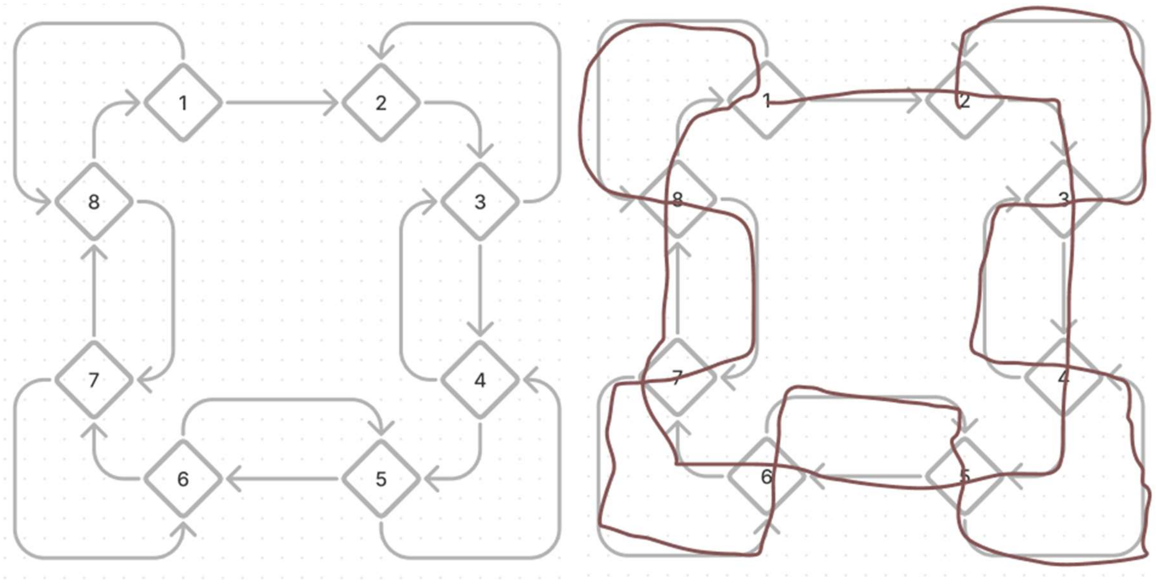


The reason why we cannot create 3 connected components with 12 vertices and 66 edges is that we cannot squeeze that much path for three separated graph. The biggest edges that 12 vertices can handle should be 36 edges.

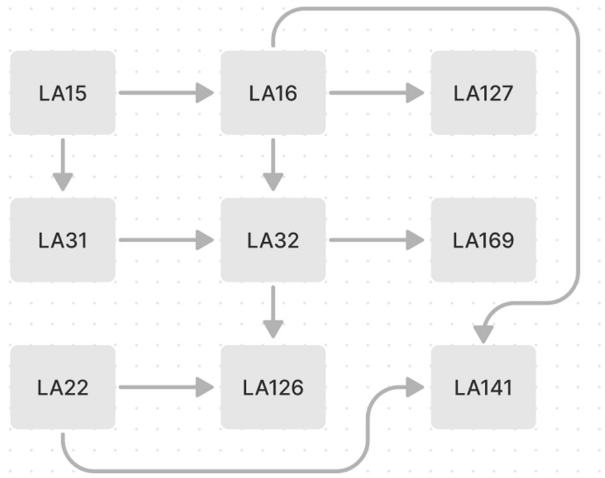
- R-13.3



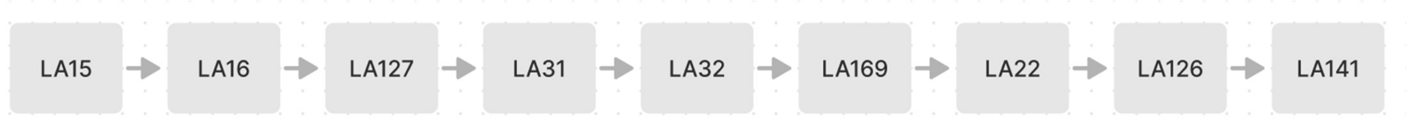
- R-13.4



- R-13.5



The sequence is:

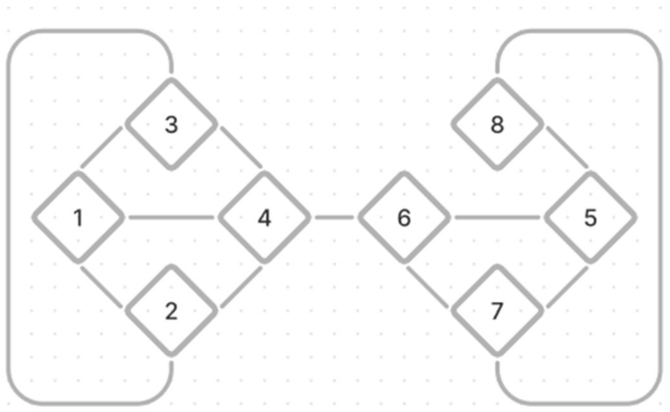


- R-13.6

If we are using edge list structure, then we do not need to manage vertex pretty much. All we need is just store the vertex data in a list. Because of that, if we are using vector for vertex holder, then the insertVertex() function will cost O(1) time. The eraseEdge() function also follow the erase operation for a list. Since we have m edges, if we find the edge is at last, then we have to cost O(m) time to do that.

- R-13.7

a.



b. (Smaller number first) 1, 2, 4, 3, 6, 7, 5, 8

c. (Smaller number first) 1, 2, 3, 4, 6, 5, 7, 8

- R-13.8

a. We have to use adjacency list structure since it asked us to use as little space as possible. There are only 20,000 edges, and if we use adjacency matrix structure, we will have 100,000,000 – 20,000 = 99,980,000 unused spaces, which is a large amount of waste.

b. We have to use adjacency list structure since it asked us to use as little space as possible. There are only 20,000,000 edges, and if we use adjacency matrix structure, we will have 100,000,000 – 20,000,000 = 80,000,000 unused spaces, which is a large amount of waste.

c. We have to use adjacency matrix structure since it asked us to do the isAdjacentTo function in O(1) time, which is only possible for matrix. We can search the value by vertex a as a row index and vertex b as a column index, and see if there is a value inside. If yes, it is adjacent. If no, it isn't.

- R-13.11

(Using DFS algorithm, start first at BOS, travel downward vertex first, start second at ORD)

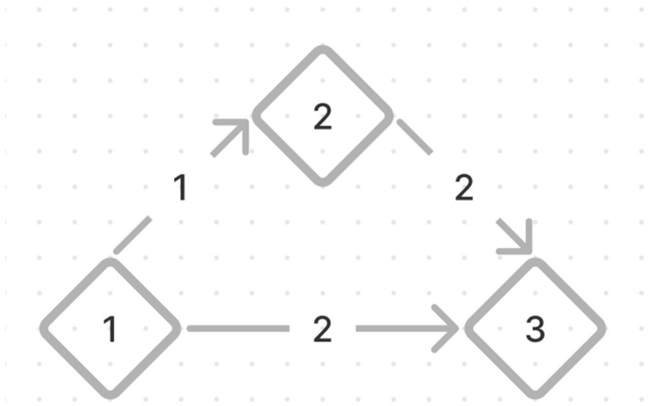
ORD, BOS, JFK, MIA, DFW, SFO, LAX

- R-13.29

It should have n – 1 edges so that its transitive closure consist of a simple directed path.

- C-13.16

It does not always find the shortest path, here is the example:



If we use the greedy strategy showed in question, it will give us the shortest path as (1, 2, 3) with weight 3. However, the shortest path should be (1, 3) with weight 2.

- C-13.21

Based on the graph G that represents the NASA station relationship, we use either Kruskal's algorithm or the Prim-Jarnik algorithm to generate the Minimum Spanning Tree. This should cost us O(mlongn) time whatever the algorithm we are using where m is the number of edges and n is the number of vertices.

Chapter 14

- R-14.6

Here is the process visualization with sequence (2, 3, 4, 1, 2, 5, 1, 3, 5, 4, 1, 2, 3), highlighted part means something has been thrown during the process:

						3		4		2		3	5
				2	3	4	4	2	2	1	3	5	4
			2	3	4	1	2	5	1	3	5	4	1
		2	3	4	1	2	5	1	3	5	4	1	2
	2	3	4	1	2	5	1	3	5	4	1	2	3
^2	^3	^4	^1	^2	^5	^1	^3	^5	^4	^1	^2	^3	

As a result, the total page missed for LRU is: 4

- R-14.7

Here is the process visualization with sequence (2, 3, 4, 1, 2, 5, 1, 3, 5, 4, 1, 2, 3), highlighted part means something has been thrown during the process:

					2	3	4	1	2	5	1	3	5
				2	3	4	1	2	5	1	3	5	4
			2	3	4	1	2	5	1	3	5	4	1
		2	3	4	1	2	5	1	3	5	4	1	2
	2	3	4	1	2	5	1	3	5	4	1	2	3
^2	^3	^4	^1	^2	^5	^1	^3	^5	^4	^1	^2	^3	

As a result, the total page missed for FIFO is: 9