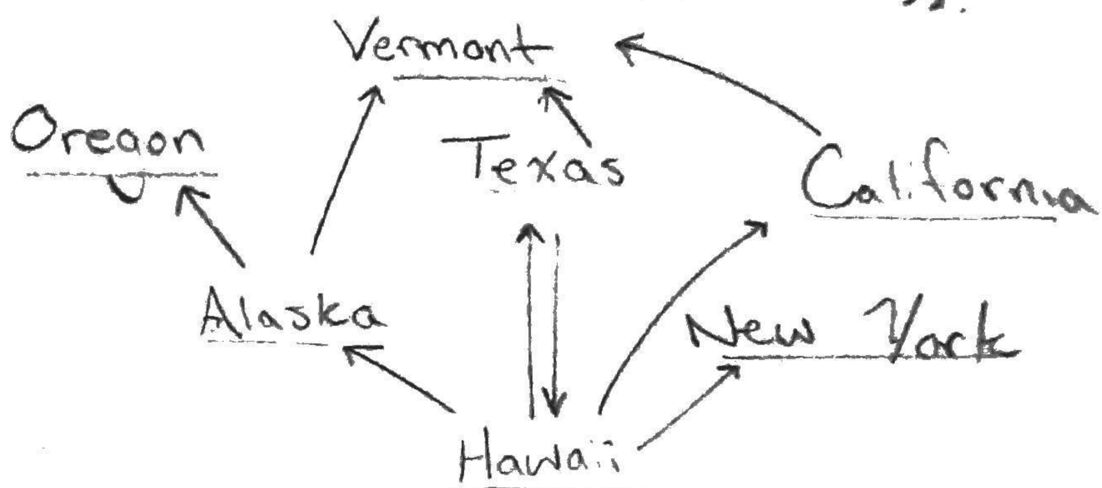
Graph Lab, CMSC 204
due 11:59pm @ Dec. 5 2020
Lucas Sicard

1. Draw the State Graph

V(StateGraph) = (Oregon, Alaska, Texas, Hawaii, Vermont, New York, California)

E(StateGraph) = [(Alaska, Oregon), (Hawaii, Alaska), (Hawaii, Texas), (Texas, Hawaii), (Hawaii, Cali.), (Hawaii, New York), (Texas, Vermont), (Vermont, Cali.), (Vermont, Alaska)].



3a. Adjacency Matrix

(u) (v)	AL	CA	HI	NY	OR	TX	VT
Alaska	Ø	Ø	Ø	Ø	1	Ø	1
Cali.	Ø	Ø	Ø	Ø	Ø	Ø	1
Hawaii	1	1	Ø	1	Ø	1	Ø
New York	Ø	Ø	Ø	Ø	Ø	Ø	Ø
Oregon	Ø	Ø	Ø	Ø	Ø	Ø	Ø
Texas	Ø	Ø	1	Ø	Ø	Ø	1
Vermont	Ø	Ø	Ø	Ø	Ø	Ø	Ø

3b. Adjacency List

Alaska → [Oregon/] → [Vermont/]

Cali. → [Vermont/]

Hawaii → [Alaska/] → [Cali./] → [New York/] → [Texas/]

New York

Oregon

Texas → [Hawaii/] → [Vermont/]

Vermont

4a. D

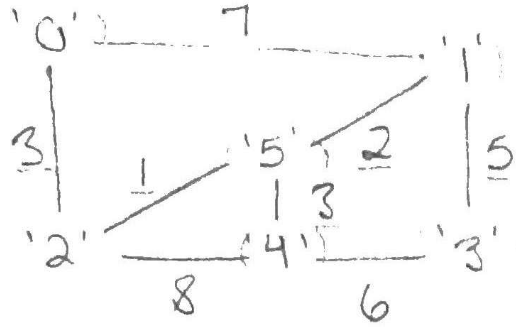
4b. A

5. Shortest distance from Altanta to every other city?

	Atlanta	Austin	Chicago	Dallas	Denver	Houston	W.
Atlanta	0	2100	2800	1900	2680	800	600

6 Find minimal spanning tree via Primm's algorithm. Start at 0 as source vertex.

Initial



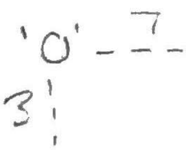
Primm's algorithm

Take the minimum cost at each vertex while maintaining the tree

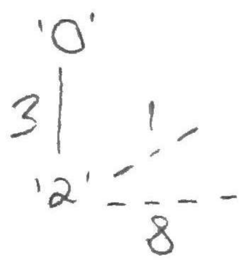
$V = \{0, 1, 2, 3, 4, 5\}$ $|V| = 6$

$E \; \{(0,1), (0,2), \ldots\}$ $|E| = |V| - 1 = 5$

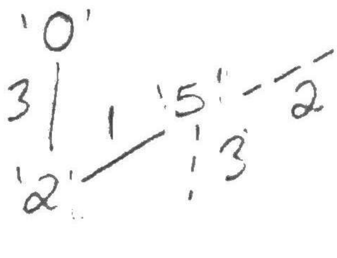
Primm's Spanning Tree

① 

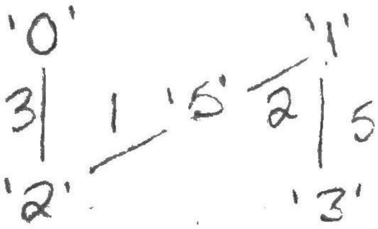
3<7

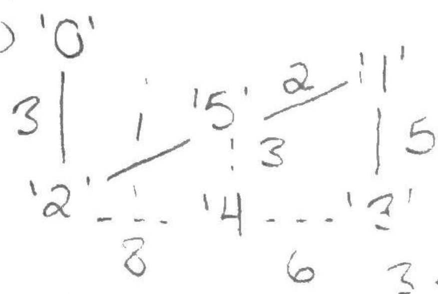
② 

1<8

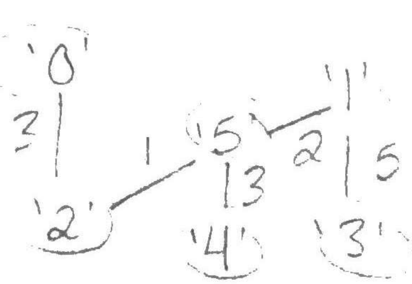
③ 

2 < 3

④ 

⑤ 

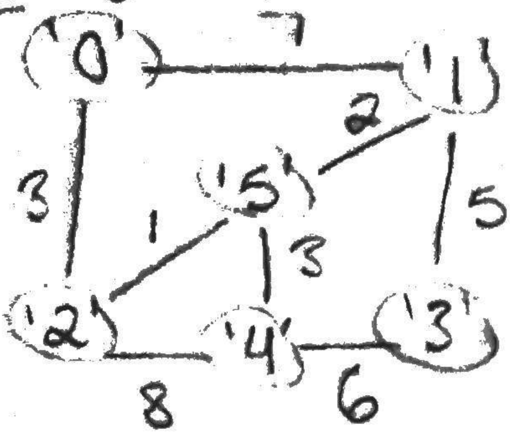
3 < 6 < 5

⑥ 

Spanning Tree Cost: 14

7. Find minimum spanning tree using Kruskal's algorithm

Initial



Kruskal's algorithm

Always select the minimum cost edge, but maintain the tree.

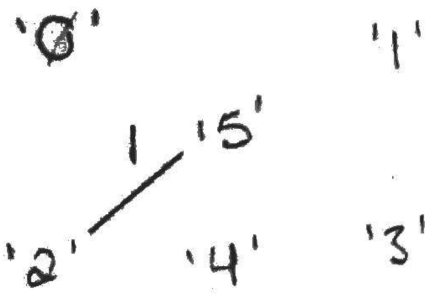
$O(|V||E|) = O(n^2)$

$|V| = 6$

$|E| = |V| - 1 = 5$

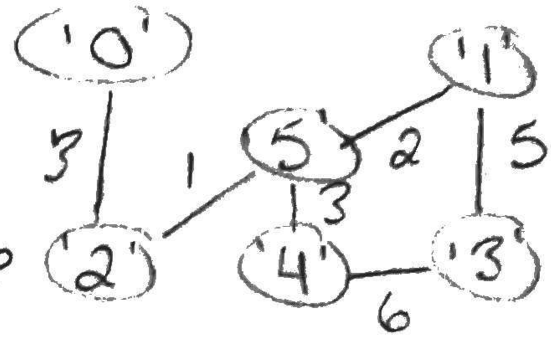
Kruskal's Minimum Spanning Tree

①



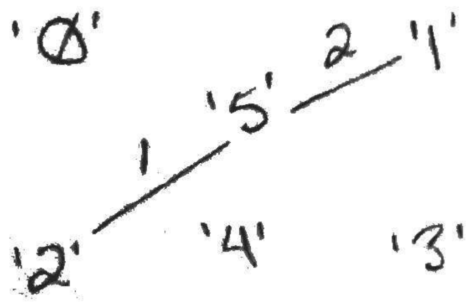
min. edge: 1

min. edge: 6



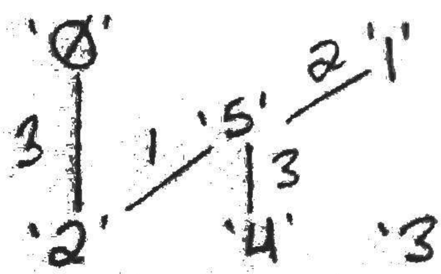
Spanning Tree Cost: 14

②



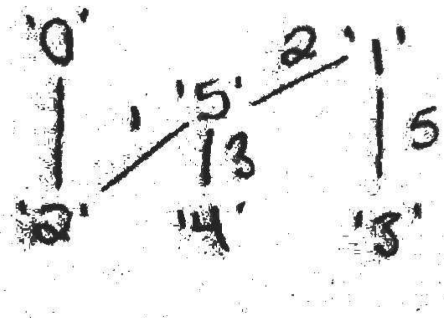
min. edge: 2

③

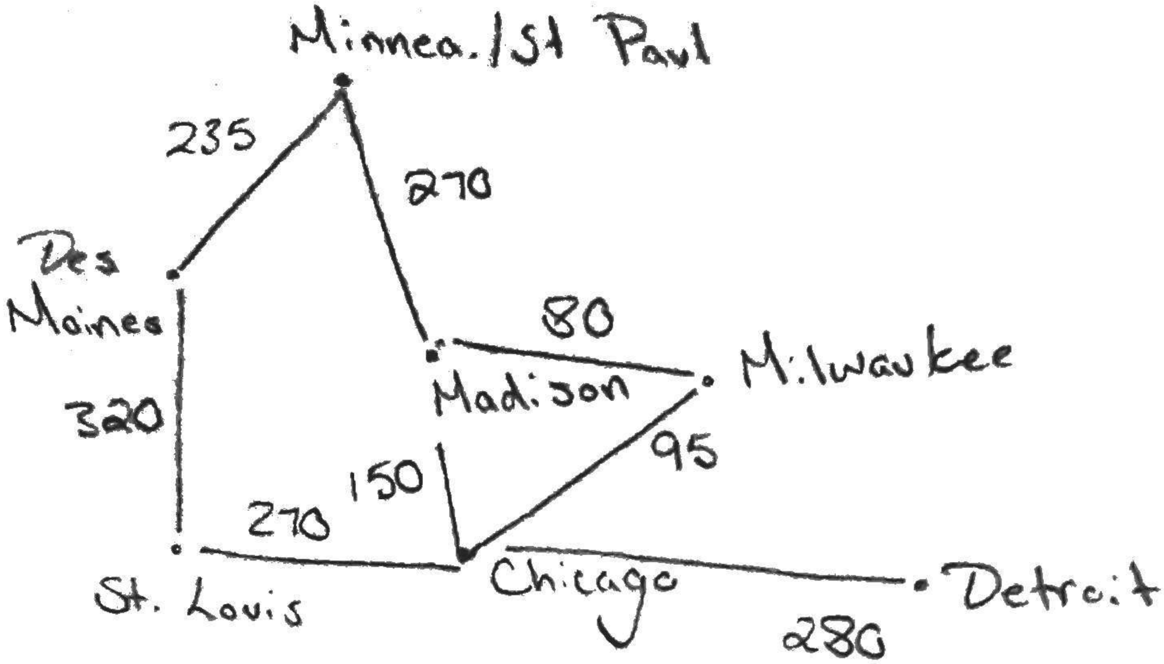


min. edge: 3

④

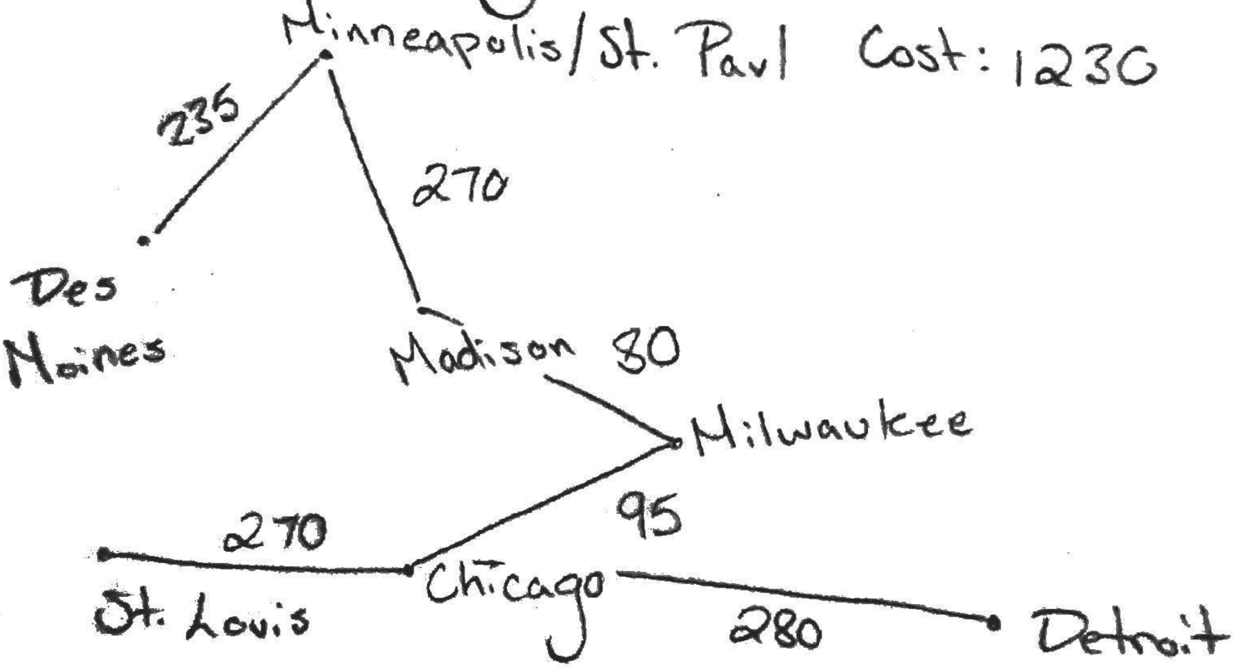


8 Find the minimum spanning tree via preferred method. Use Minneapolis/St. Paul as source vertex.

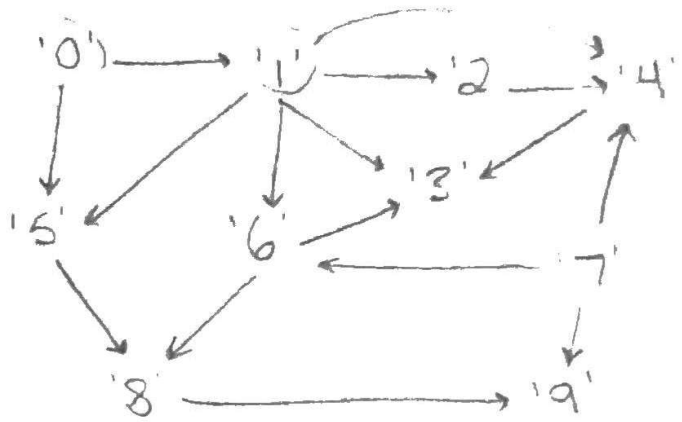


Minimum Spanning Tree:

Cost: 1230



9. List the nodes of the graph in a breadth-first ordering. Show the steps using arrays predCount, topologicalOrder, and a queue.



Proof

Step	Queue	predCount	topologicalOrder
1 start at '0'	0 $\xrightarrow{\text{pop}}$	0	$\xrightarrow{\text{output}}$ 0
2 get children of '0'	51 (front) $\xrightarrow{\text{pop '1'}}$	01	$\xrightarrow{\text{out}}$ 01
3 get child of '5'	654325 $\xrightarrow{\text{pop '5'}}$	015	$\xrightarrow{\text{out}}$ 015
4 get child of '2'	865432 $\xrightarrow{\text{pop '2'}}$	0152	$\xrightarrow{\text{out}}$ 0152
5 get child of '3'	486543 $\xrightarrow{\text{pop '3'}}$	01523	$\xrightarrow{\text{out}}$ 01523
6 get child of '4'	348654 $\xrightarrow{\text{pop '4'}}$	015234	$\xrightarrow{\text{out}}$ 015234
7 '5' visited	34865	015234	015234
8 get child of '6'	8386 $\xrightarrow{\text{pop '6'}}$	0152346	$\xrightarrow{\text{out}}$ 0152346
9 get child of '8'	9838 $\xrightarrow{\text{pop '8'}}$	01523468	$\rightarrow$ 01523468
10 '3' visited	983	01523468	01523468

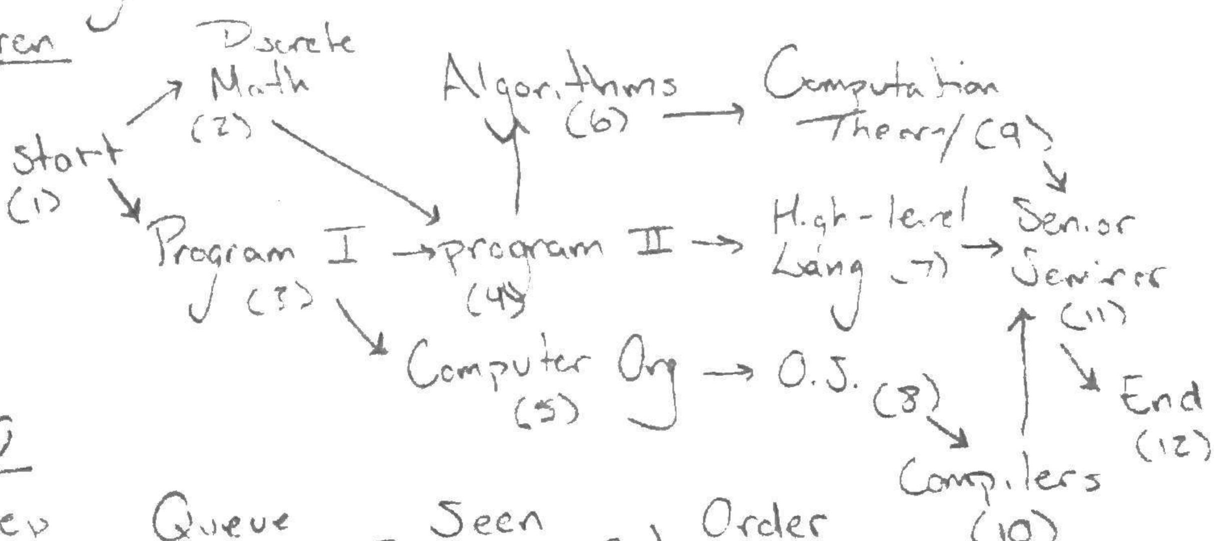
		Queue	predCount	topologicalOrder
(11)	'8' visited	9 ~~8~~	01523468	01523468
(12)	No children of '9'	9 $\xrightarrow{\text{pop '9'}}$	015234689	→ 015234689

Node '7' neighbors nodes '4', '6', and '9'.
Node '7' is a neighbor to none however

Breadth First topological order:
015234689

10. List the nodes of the graph in a breadth first topological order

Given



BFO

Step	Queue		Seen		Order
1.	1	$\xrightarrow{\text{pop}}$	1	$\xrightarrow{\text{out}}$	1
2.	432	→	12	→	12
3.	5443	→	123	→	123
4.	765~~1~~4	→	1234	→	1234
5.	8765	→	12345	→	12345

Step	Queue		Seen		Order
6.	9876	→	123456	→	123456
7.	(11)987	→	1234567	→	1234567
8.	(10)(11)98	→	12345678	→	12345678
9.	(11)(10)(11)9	→	123456789	→	123456789
10.	(12)(11)(10)(11)	→	123456789(11)	→	123456789(11)
11.	~~(11)~~(12)~~(11)~~(10)	→	123456789(11)(10)	→	123456789(11)(10)
12	(12)	→	123456789(11)(10)(12)	→	123456789(11)(10)(12)

Breadth First Order:

Start, Discrete Math, Program I, Program II, Computer Organization, Algorithms, High Level Languages, Operating Systems, Theory of Computation, Senior Seminar, Compilers, End.