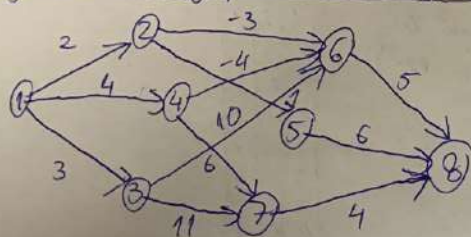


Topological sort using predecessor counting algorithm.

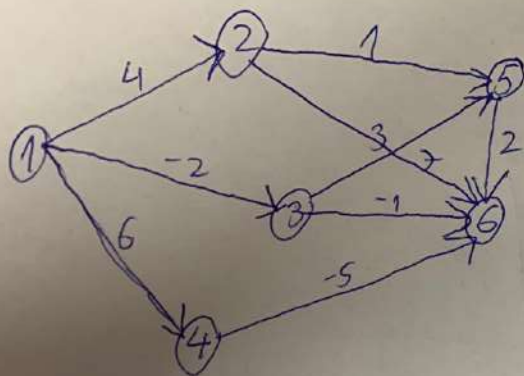


Input file:
("put me")

1	2	2
1	4	4
1	3	3
2	5	1
2	6	-3
3	6	10
3	7	11
4	6	-4
4	7	6
5	8	6
6	8	5
7	8	4

BLOG RICCARDO
911

	current vertex	neighbour	count: dict	q: queue	sorted-list: list																					
initialization			<table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>3</td><td>2</td><td>3</td></tr></table>	1	2	3	4	5	6	7	8	0	1	1	1	1	3	2	3	<table><tr><td>←</td><td>1</td><td>←</td></tr></table>	←	1	←	[]		
1	2	3	4	5	6	7	8																			
0	1	1	1	1	3	2	3																			
←	1	←																								
iteration 1	1	2 4 3	<table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>3</td><td>2</td><td>3</td></tr></table>	1	2	3	4	5	6	7	8	0	0	0	0	1	3	2	3	<table><tr><td>←</td><td>2</td><td>4</td><td>3</td><td>←</td></tr></table>	←	2	4	3	←	[1]
1	2	3	4	5	6	7	8																			
0	0	0	0	1	3	2	3																			
←	2	4	3	←																						
iteration 2	2	5 6	<table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>2</td><td>2</td><td>3</td></tr></table>	1	2	3	4	5	6	7	8	0	0	0	0	0	2	2	3	<table><tr><td>←</td><td>4</td><td>3</td><td>5</td><td>←</td></tr></table>	←	4	3	5	←	[1, 2]
1	2	3	4	5	6	7	8																			
0	0	0	0	0	2	2	3																			
←	4	3	5	←																						
iteration 3	4	6 7	<table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>3</td></tr></table>	1	2	3	4	5	6	7	8	0	0	0	0	0	1	1	3	<table><tr><td>←</td><td>3</td><td>5</td><td>←</td></tr></table>	←	3	5	←	[1, 2, 4]	
1	2	3	4	5	6	7	8																			
0	0	0	0	0	1	1	3																			
←	3	5	←																							
iteration 4	3	6 7	<table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3</td></tr></table>	1	2	3	4	5	6	7	8	0	0	0	0	0	0	0	3	<table><tr><td>←</td><td>5</td><td>6</td><td>7</td><td>←</td></tr></table>	←	5	6	7	←	[1, 2, 4, 3]
1	2	3	4	5	6	7	8																			
0	0	0	0	0	0	0	3																			
←	5	6	7	←																						
iteration 5	5	8	<table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr><tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td><td>-1</td><td>-1</td><td>-1</td><td>2</td></tr></table>	1	2	3	4	5	6	7	8	-1	-1	-1	-1	-1	-1	-1	2	<table><tr><td>←</td><td>6</td><td>7</td><td>←</td></tr></table>	←	6	7	←	[1, 2, 4, 3, 5]	
1	2	3	4	5	6	7	8																			
-1	-1	-1	-1	-1	-1	-1	2																			
←	6	7	←																							
iteration 6	6	8	<table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr><tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td><td>-1</td><td>-1</td><td>-1</td><td>1</td></tr></table>	1	2	3	4	5	6	7	8	-1	-1	-1	-1	-1	-1	-1	1	<table><tr><td>←</td><td>7</td><td>←</td></tr></table>	←	7	←	[1, 2, 4, 3, 5, 6]		
1	2	3	4	5	6	7	8																			
-1	-1	-1	-1	-1	-1	-1	1																			
←	7	←																								
iteration 7	7	8	<table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	1	2	3	4	5	6	7	8	0	0	0	0	0	0	0	0	<table><tr><td>←</td><td>8</td><td>←</td></tr></table>	←	8	←	[1, 2, 4, 3, 5, 6, 7]		
1	2	3	4	5	6	7	8																			
0	0	0	0	0	0	0	0																			
←	8	←																								
								[1, 2, 4, 3, 5, 6, 7, 8]																		



Topological orders:

1. 1, 2, 3, 4, 5, 6
2. 1, 3, 2, 4, 5, 6
3. 1, 2, 4, 3, 5, 6
4. 1, 3, 4, 2, 5, 6
5. 1, 4, 2, 3, 5, 6
6. 1, 4, 3, 2, 5, 6

Input file:
("puzzle")

1	2	4
1	3	-2
1	4	6
2	5	1
2	6	7
3	5	3
3	6	-1
4	6	-5
5	6	2

BLOG RICCARDO

From 1 to 6: highest cost path = [1, 2, 6], cost = 11

From 1 to 5: highest cost path = [1, 2, 5], cost = 5

From 3 to 6: highest cost path = [3, 5, 6], cost = 5