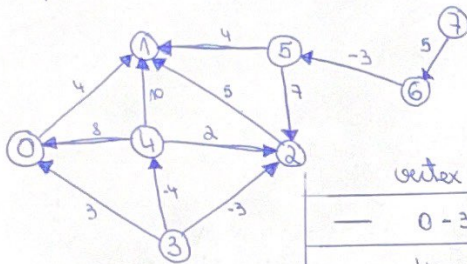


Graph :



Input file

8	12	4
0	1	5
2	1	-4
3	2	-3
3	0	3
4	0	8
4	1	10
4	2	2
5	1	4
5	2	7
6	5	-3
7	6	5

highest cost path

Manual execution

Topological sort

sorted-v = []
fully processed = empty set
in-process-v = empty set

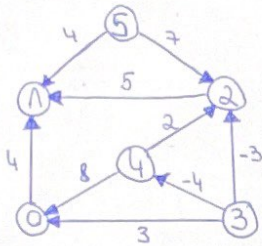
vertex	sorted-v	fully processed	in-process	
— 0-3	[3]	{3}	{0}	returns true
4	[3,4]	{3,4}	{0}	returns true
0	[3,4,0]	{0,3,4}	{ }	returns true
— 1-7	[3,4,0,7]	{0,3,4,7}	{1,2,5,6}	returns true
6	[3,4,0,7,6]	{0,3,4,6,7}	{1,2,5}	returns true
5	[3,4,0,7,6,5]	{0,3,4,5,6,7}	{1,2}	returns true
2	[3,4,0,7,6,5,2]	{0,2,3,4,5,6,7}	{1}	returns true
1	[3,4,0,7,6,5,2,1]	{0,1,2,3,4,5,6,7}	{ }	returns true

vertex	cost list [-∞ × multiplicity, 0]	previous list [None × multiplicity]			
3	[-inf, -inf, -inf, -inf, -inf, 0]	[None, ..., None]	6	[inf, ..., 2, 5, 0]	[None, ..., 6, 7, None]
4	— —	— —	5	[inf, 6, 9, -inf, -inf, 2, 5, 0]	[None, 5, 5, None, 1, 6, 7, None]
0	— —	— —	2	[inf, 11, 3, -inf, -inf, 2, 5, 0]	[1, 2, 5, 11, 11, 6, 7, None]
7	[-inf, -inf, -inf, -inf, -inf, 5, 0]	[None, ..., 7, None]	1	— —	— —

The reconstructed path: [7, 6, 5, 2, 1]

maximum cost 14

Graph



input file

```
8 9
0 1 4
2 1 5
3 4 -4
3 2 -3
3 0 3
4 0 8
4 2 2
5 1 7
5 2 7
```

Highest path cost path between :

0,1 - [0,1] of 4
0,2 - no path
0,3 - no path
0,4 - no path
0,5 - no path

1,0 - no path
1,2 - no path
1,3 - no path
1,4 - no path
1,5 - no path

2,0 - no path
2,1 - [2,1] of 5
2,3 - no path
2,4 - no path
2,5 - no path

3,0 - [3,0] of 4
3,1 - [3,1] of 8
3,2 - [3,2] of -2
3,4 - [3,4] of -4
3,5 - no path

4,0 - [4,0] of 8
4,1 - [4,1] of 12
4,2 - [4,2] of 2
4,3 - no path
4,5 - no path

5,0 - no path
5,1 - [5,1] of 12
5,2 - [5,2] of 7
5,3 - no path
5,4 - no path