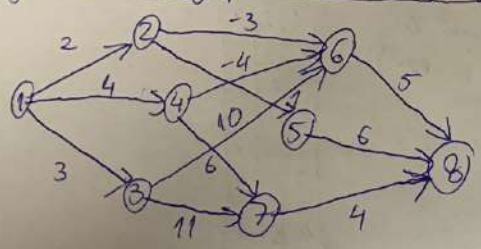
# Topological sort using predecessor counting algorithm:

BOLOG RICCARDO
911



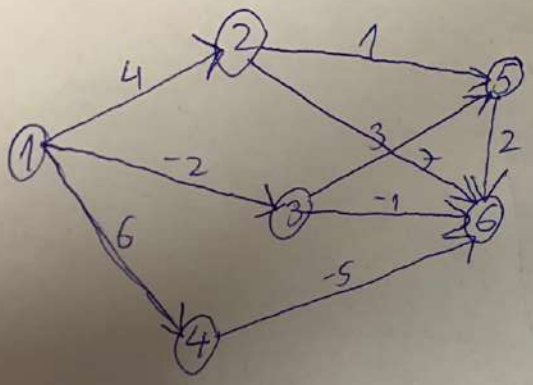

Input file:
("p4.txt")

| 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 |

| | current_vertex | neighbour | count: dict | q: queue | sorted_list: list |
|---|---|---|---|---|---|
| initialization | | | 1 2 3 4 5 6 7 8<br>0 1 1 1 1 3 2 3 | ← 1 ← | [ ] |
| iteration 1 | 1 | 2<br>4<br>3 | 1 2 3 4 5 6 7 8<br>0 0 0 0 1 3 2 3 | ← 2 4 3 ← | [1] |
| iteration 2 | 2 | 5<br>6 | 1 2 3 4 5 6 7 8<br>0 0 0 0 0 2 2 3 | ← 4 3 5 ← | [1,2] |
| iteration 3 | 4 | 6<br>7 | 1 2 3 4 5 6 7 8<br>0 0 0 0 0 1 1 3 | ← 3 5 ← | [1,2,4] |
| iteration 4 | 3 | 6<br>7 | 1 2 3 4 5 6 7 8<br>0 0 0 0 0 0 0 3 | ← 5 6 7 ← | [1,2,4,3] |
| iteration 5 | 5 | 8 | —11— 8 2 | ← 6 7 ← | [1,2,4,3,5] |
| iteration 6 | 6 | 8 | —11— 8 1 | ← 7 ← | [1,2,4,3,5,6] |
| iteration 7 | 7 | 8 | 1 2 3 4 5 6 7 8<br>0 0 0 0 0 0 0 0 | ← 8 ← | [1,2,4,3,5,6,7] |
| | | | | | [1,2,4,3,5,6,7,8] |

BOLOG RICCARDO




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 ("nuovo2"):

| | | |
|---|---|---|
| 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 |

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