TOPIC

You are going to simulate a smart robot who aims to collect artifacts in a labyrinth with at most 10 rooms, including the entrance room. Each room can have at most 10 artifacts except the entrance room, which is empty. Each artifact has a weight (an integer number greater than 0kg) and a value (an integer amount greater than 0 TL). Since the artifacts are indivisible, the robot should either take it or leave it (*Hint*: *1-0 Knapsack Problem*). Unfortunately, your robot can take up to a limited weight from each room. The maximum weight that can be taken from each room is written on the rooms' doors, so your robot knows it. The robot aims to find *the longest path* in the graph starting from the entrance room to maximize its gain.

Find the longest path in the labyrinth with topological sort

Here is a sample **input.txt** structur

```
0    1    1    0    0
0    0    1    1    1
0    0    0    1    1
0    0    0    0    1
0    0    0    0    0
8
```

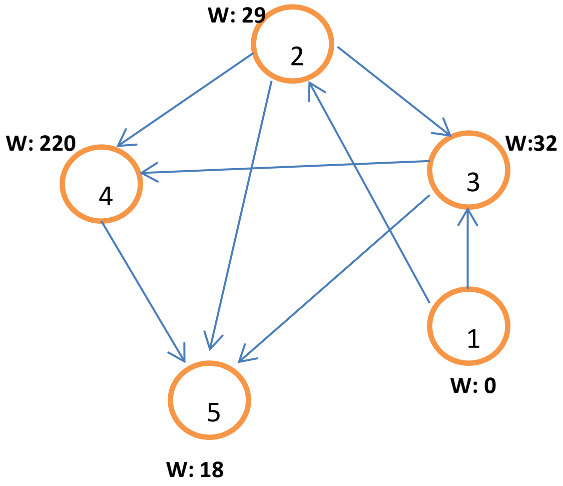
Adjacency matrix for the rooms in the labyrinth. The first room is the entrance with no artifacts.

Total weight that can be taken from the 2nd room.

1	15	Weight/Value matrix for the 2nd room.
5	10	
3	9	
4	5	
20		Total weight that can be taken from the 3rd room.
2	3	Weight/Value matrix for the 3rd room.
3	4	
4	5	
5	8	
9	10	
50		Total weight that can be taken from the 4th room.
10	60	Weight/Value matrix for the 4th room.
20	100	
30	120	
8		Total weight that can be taken from the 5th room.
4	9	Weight/Value matrix for the 5th room.
3	8	
5	10	
6	9	
2	3	

Solving the Knapsack problem will give you 29 TL, 26 TL, 220 TL and 18 TL for each room. **Note that these are not optimum solutions**.

Considering the adjacency matrix the following graph can be visualized for the labyrinth.



Output: The longest path is 1-2-3-4-5 and the total gain is 0+29+32+220+18=299TL