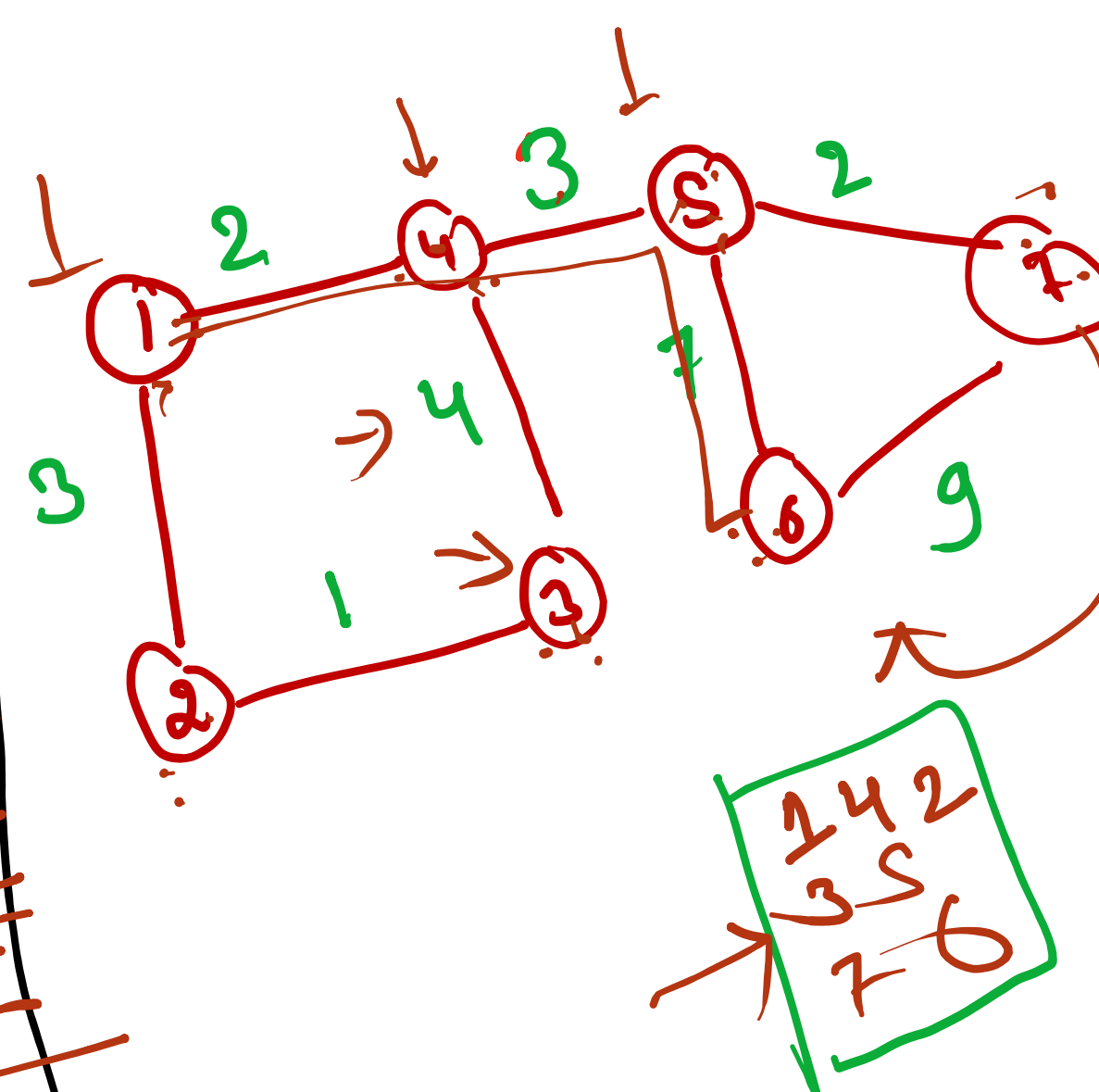


Determine Path
Disjoint
SSSP A

1 1 0 0
4 1 4 0 2
2 1 2 0 3
3 1 2 0 4
5 1 4 5 0 5
7 1 4 5 7 0 7
6 1 4 5 6 0 6

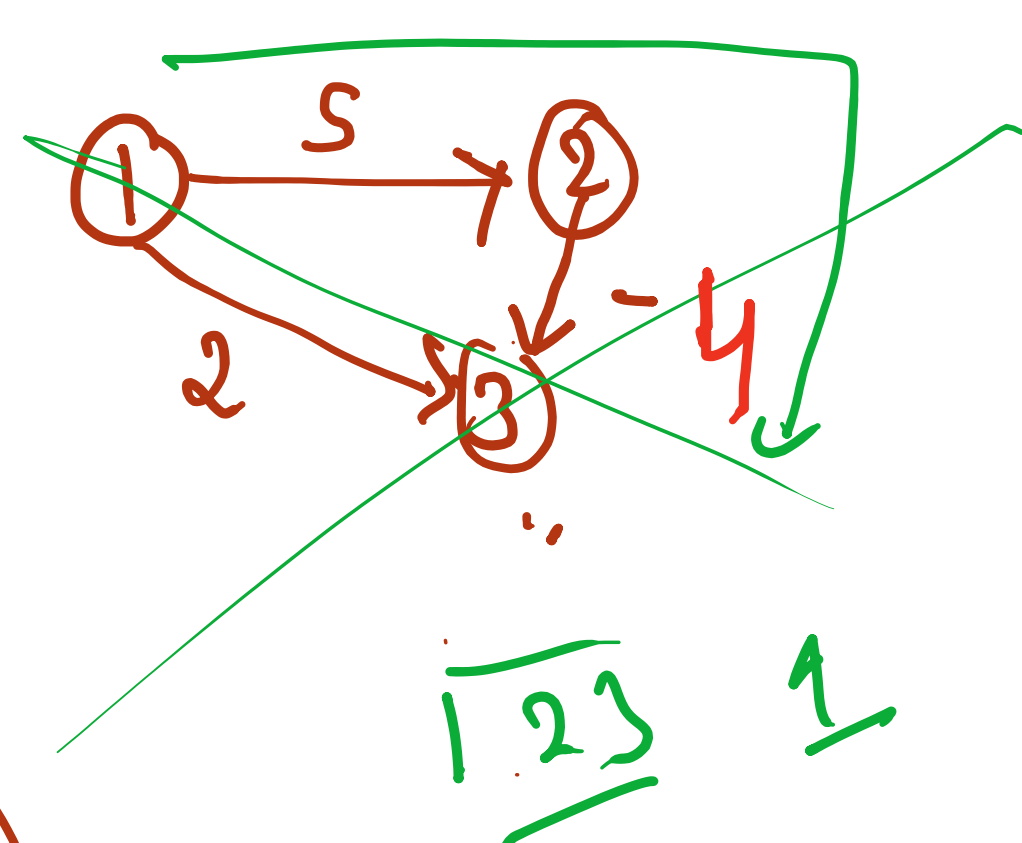
1	1	0
2	1	2
4	1	4
3	1	2
5	1	4
7	1	4
6	1	4
7	1	4
6	1	4



20
21 1-20 move
27 2-19 move
3-18 move
4-17 move
5-16 move

1-2 ✓
1-3 ✓
1-4 ✓
1-5 ✓
1-6 ✓
1-7 ✓

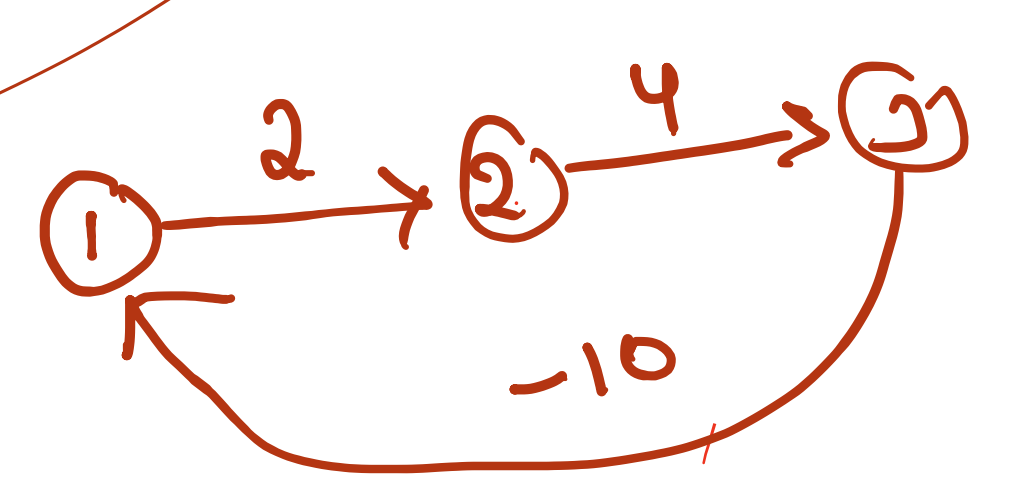
1-2
1-3
1 1 0 0
3 1 3 0 2
2 1 2 0 5



1	1	0
2	1	2
3	1	2

6
1 2 3
 $\frac{123}{6-10+2+4} = 2$

cycle 20



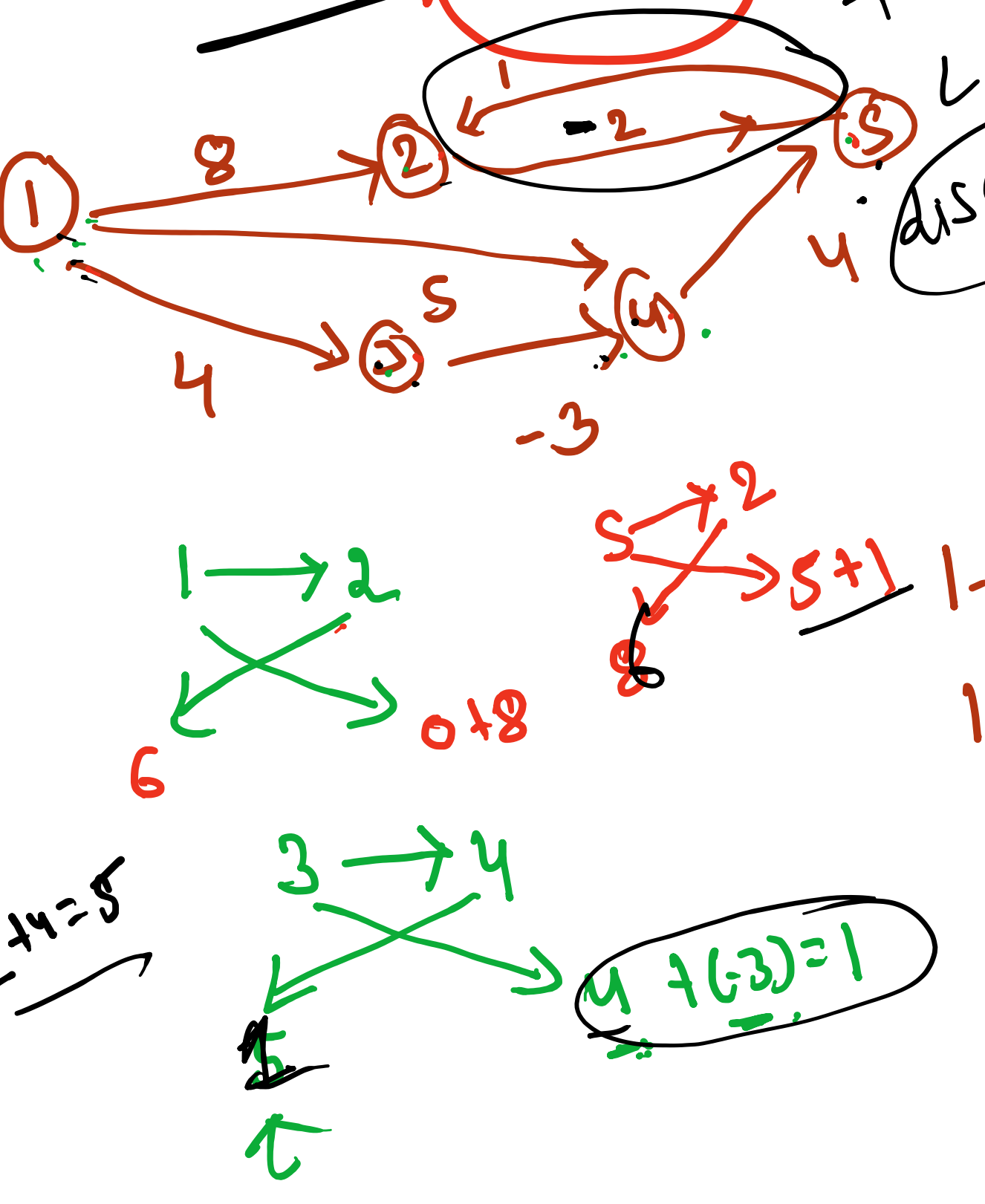
11. 200



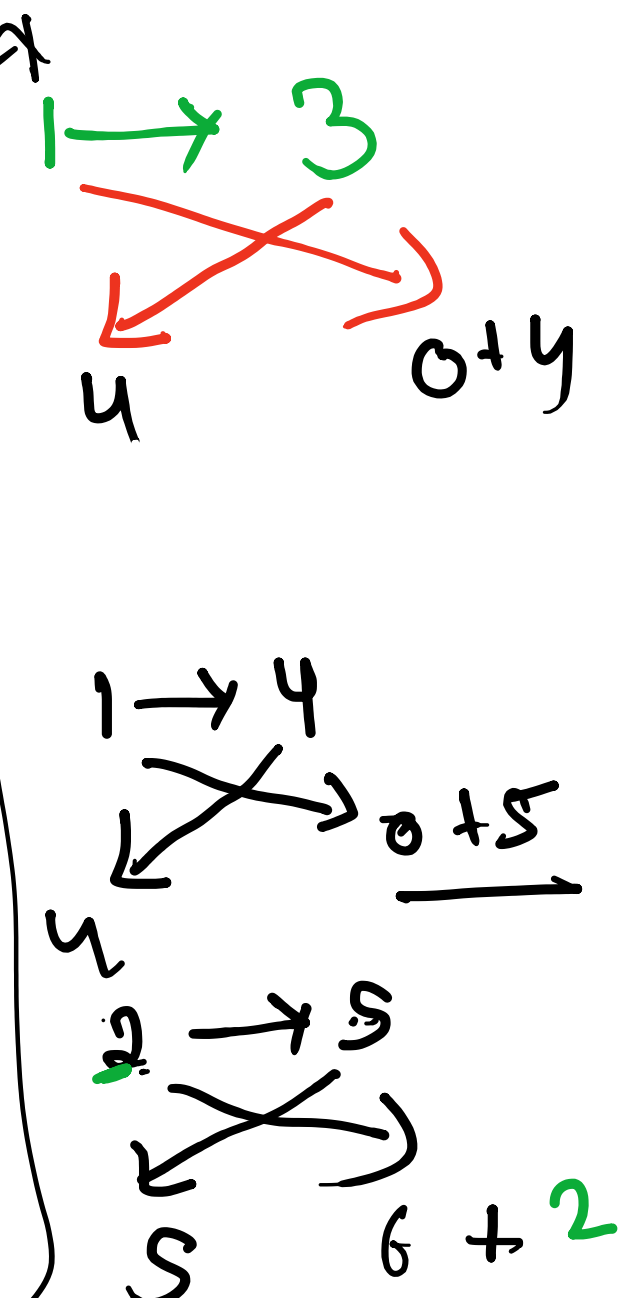
Class 2 solving 2

1	2	8
1	3	4
1	4	5
2	5	2
3	4	-3
4	5	4
5	2	4
4	5	2

disceen > directe te so



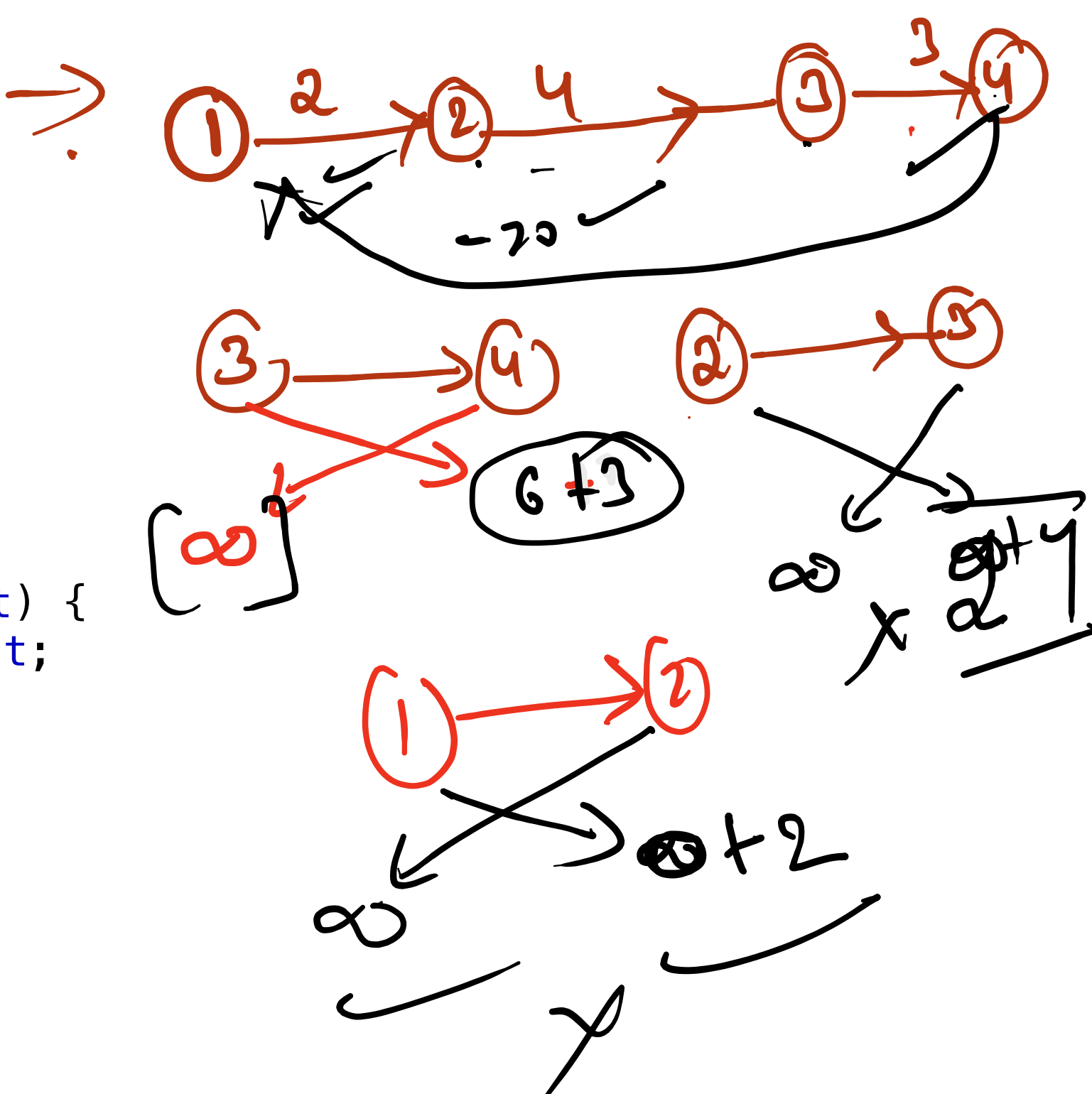
1	0
2	-6
3	4
4	1
5	5



Q1

```
public void Bellman() {  
    int v = map.size();  
    int[] dis = new int[v+1];  
    for(int i = 2; i < dis.length; i++) {  
        dis[i] = 9999999;  
    }  
    List<EdgePair> ll = GetAllEdge();  
    for(int i = 1; i < v; i++) {  
        for(EdgePair e: ll) {  
            if(dis[e.e2] > dis[e.e1] + e.cost) {  
                dis[e.e2] = dis[e.e1] + e.cost;  
            }  
        }  
    }  
    for(int i = 2; i < dis.length; i++) {  
        System.out.println(i+" "+dis[i]);  
    }  
}
```

0	2	6	9
1	2	3	4

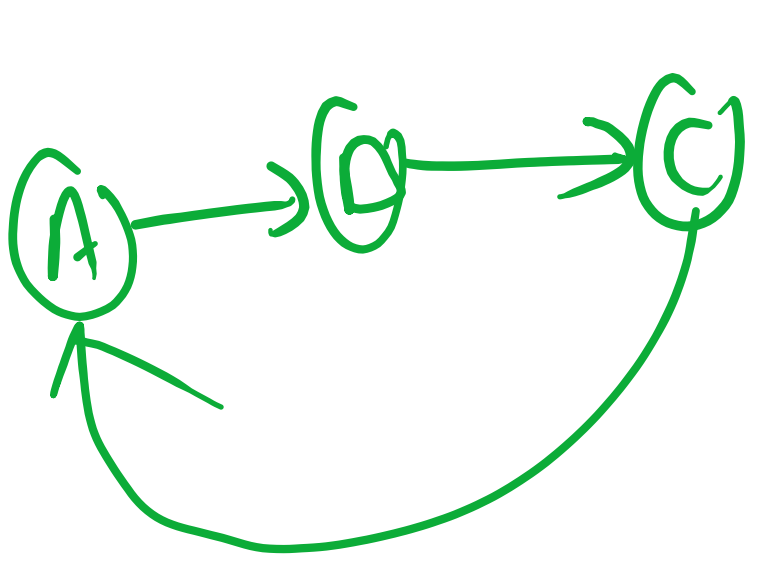
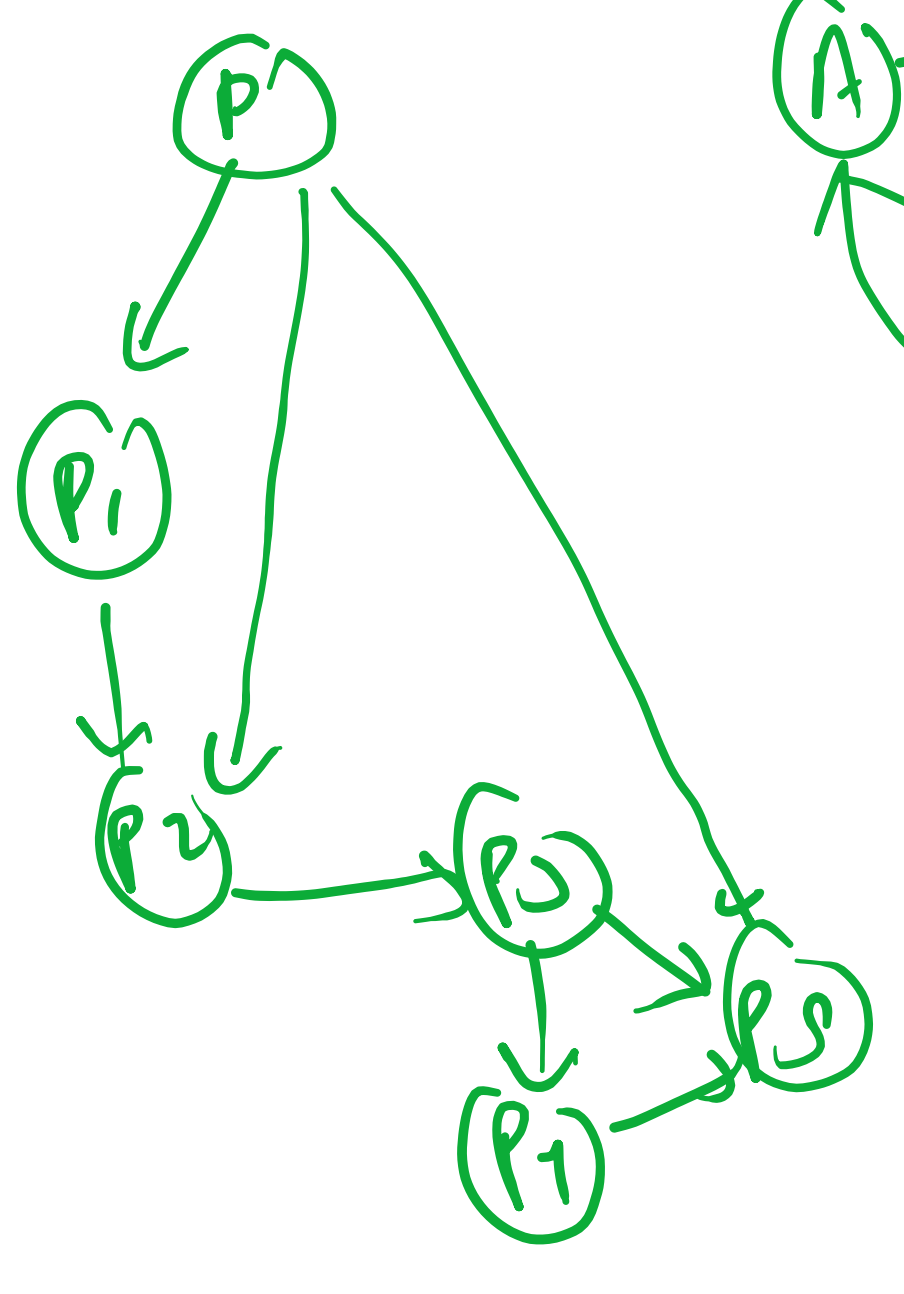
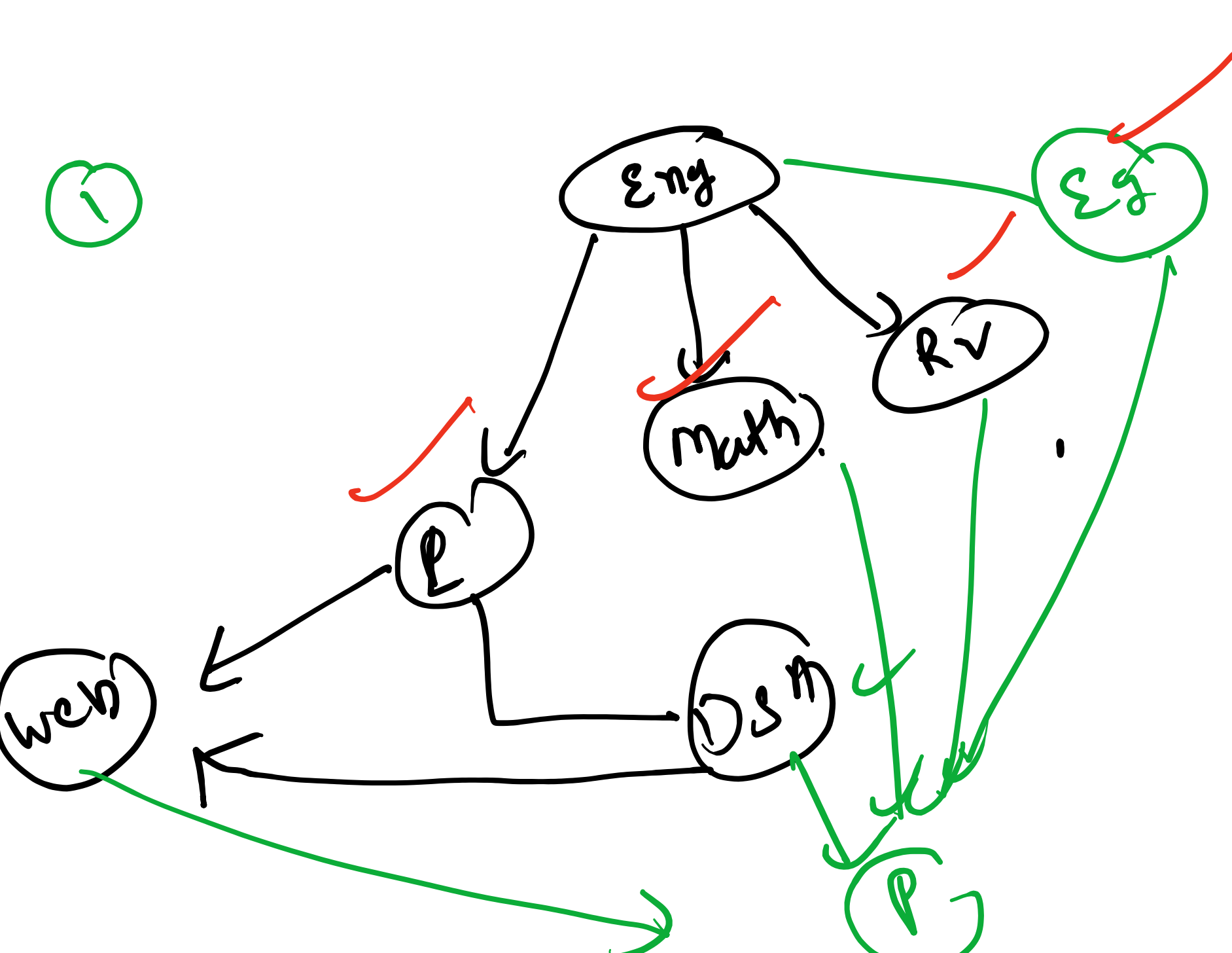
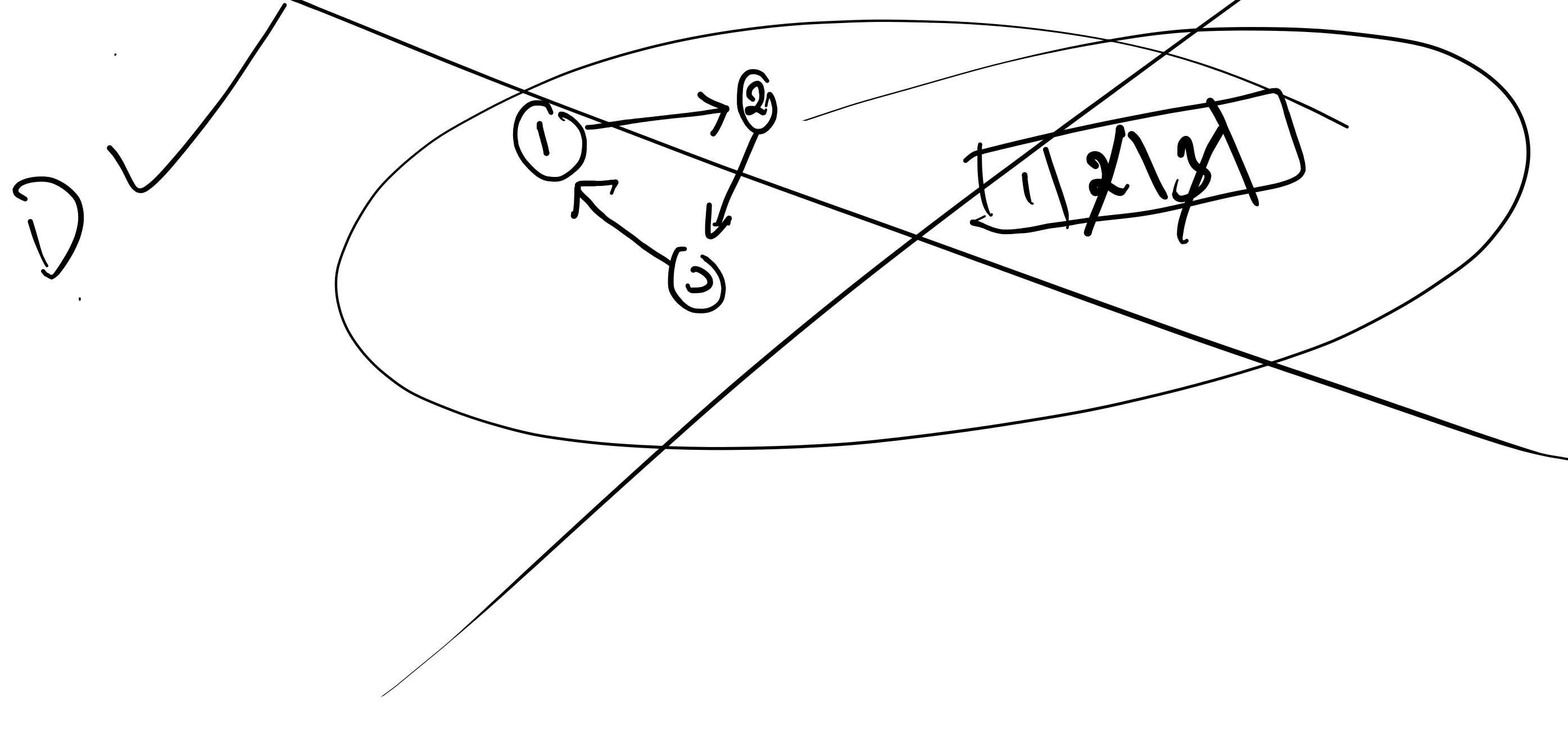


Topological

DAG

1 2 3

1 2 3

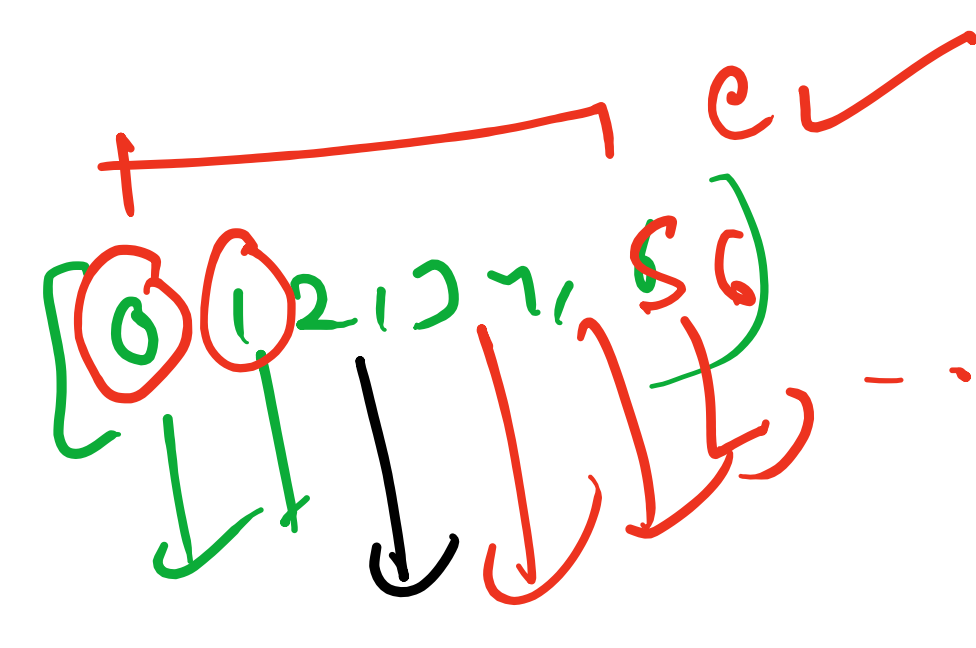


1	2	3	4	5	6
1	2	3	4	5	6

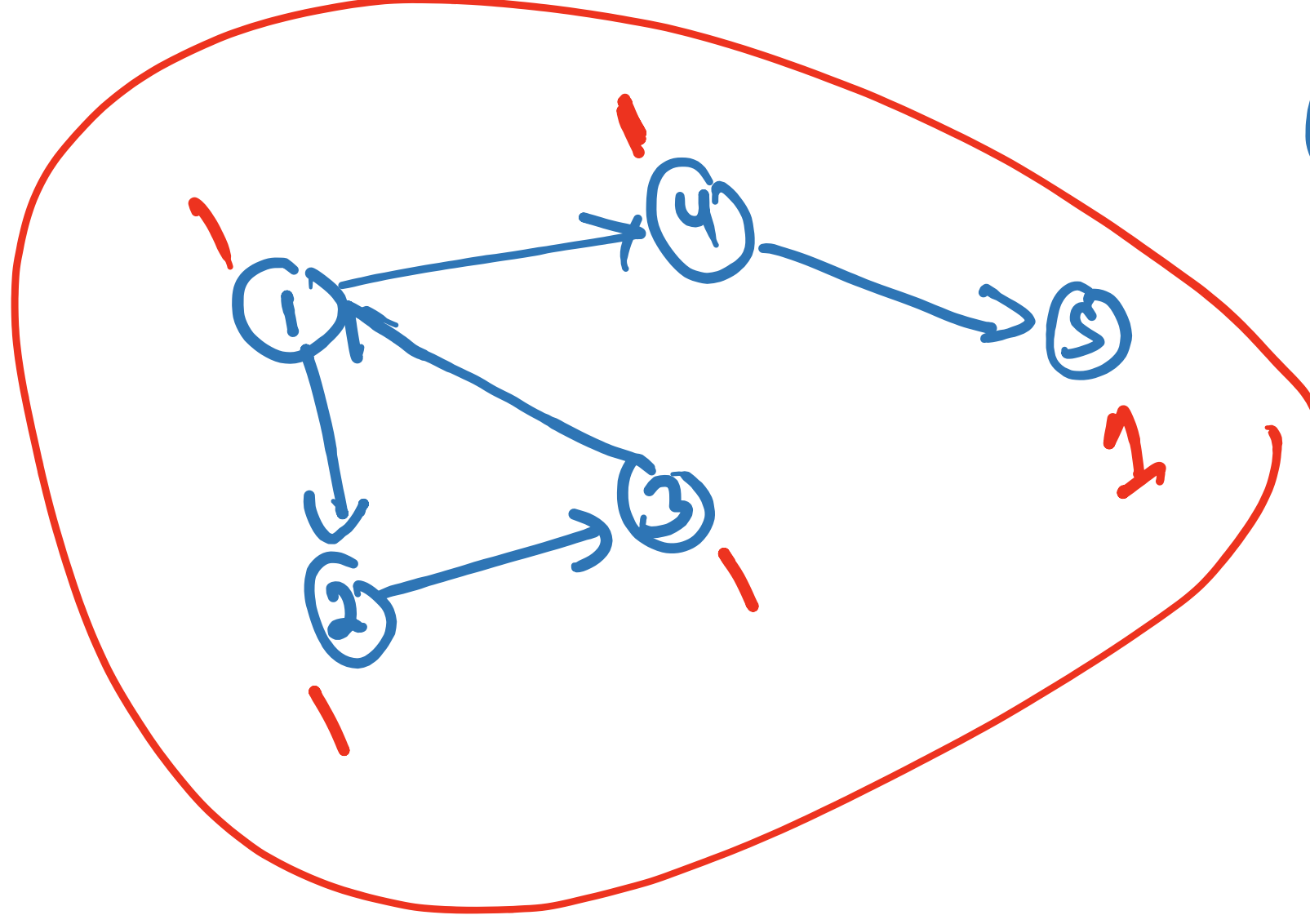
0 8
0 9
0 10
0 11
0 12
0 13
0 14
0 15
0 16
0 17
0 18
0 19
0 20
0 21
0 22
0 23
0 24
0 25
0 26
0 27
0 28
0 29
0 30
0 31
0 32
0 33
0 34
0 35
0 36
0 37
0 38
0 39
0 40
0 41
0 42
0 43
0 44
0 45
0 46
0 47
0 48
0 49
0 50
0 51
0 52
0 53
0 54
0 55
0 56
0 57
0 58
0 59
0 60
0 61
0 62
0 63
0 64
0 65
0 66
0 67
0 68
0 69
0 70
0 71
0 72
0 73
0 74
0 75
0 76
0 77
0 78
0 79
0 80
0 81
0 82
0 83
0 84
0 85
0 86
0 87
0 88
0 89
0 90
0 91
0 92
0 93
0 94
0 95
0 96
0 97
0 98
0 99

1 8 0 9 2 3 4 5 6
mat edu

1	0	1	2	1	2	1
0	1	2	3	4	5	6



6



6

