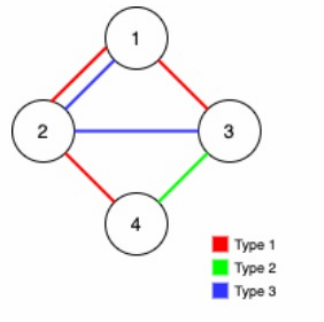
(i) priority → type 3

(ii) DSU

remove = 1 + 2



Input: n = 4, edges = [[3,1,2],[3,2,3],[1,1,3],[1,2,4],[1,1,2],[2,3,4]]

type u v

✓ 3 1 2

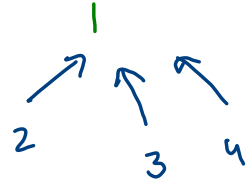
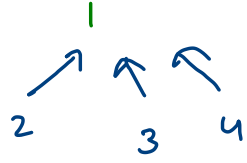
✓ 3 2 3

✗ 1 1 3

✓ 1 2 4

✗ 1 1 2

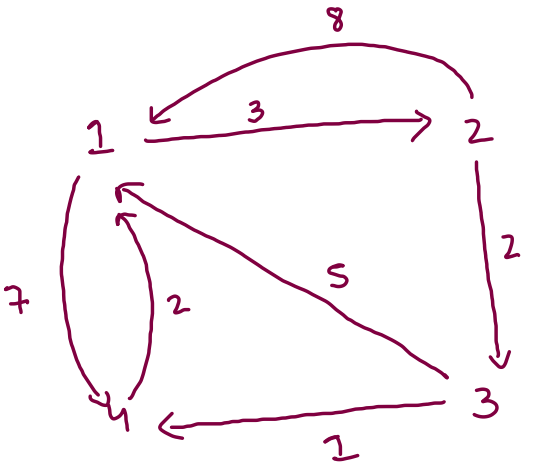
✓ 2 3 4

Alice	Bob
	
ca = 4	cb = 4

Floyd warshall : (i) all pairs shortest path.

(ii) DP based.

(iii) -ve edge wt.



Adjacency matrix

$A_0 \rightarrow$

	1	2	3	4
1	0	3	∞	7
2	8	0	2	∞
3	5	∞	0	1
4	2	∞	∞	0

$A_0 \rightarrow$

	1	2	3	4
1	0	3	∞	7
2	8	0	2	∞
3	5	∞	0	1
4	2	∞	∞	0

→ uv

no intermediate

A_1

	1	2	3	4
1	0	3	∞	7
2	8	0	2	15
3	5	8	0	1
4	2	5	∞	0

→ uv

$u i_1 v$

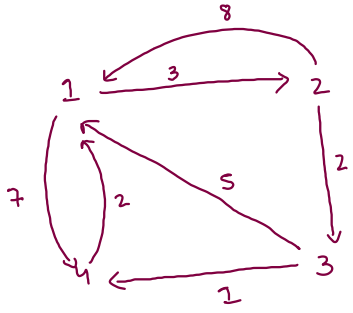
1 as intermediate

A_2

	1	2	3	4
1	0	3	5	7
2	8	0	2	15
3	5	8	0	1
4	2	5	7	0

→

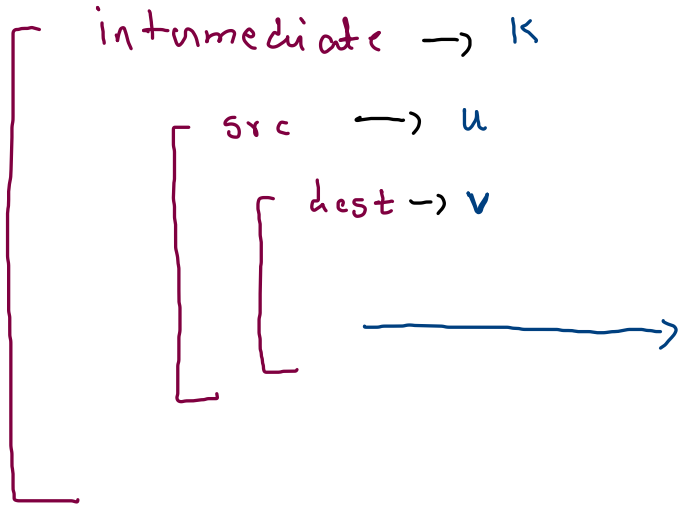
2 as intermediate



uv	(1, 2)
$u i_1 v$	(3, 2)
$u i_2 v$	(1, 3)
$u i_1 i_2 v$	(4, 3)

3 intermediate

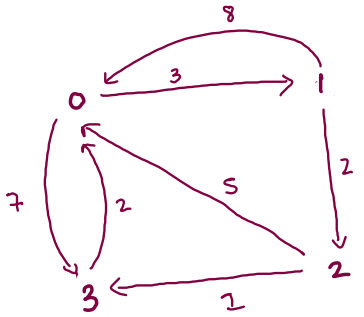
4 intermediate



$$mat[u][v] = \min \begin{pmatrix} mat[u][v], \\ mat[u][k] + mat[k][v] \end{pmatrix}$$

$A_0 \rightarrow$

	0	1	2	3
0	0	3	-1	7
1	8	0	2	-1
2	5	-1	0	1
3	2	-1	-1	0




```
//to select an indermediate
for(int k=0; k < matrix.length;k++) {
    //to select a src
    for(int u=0; u < matrix.length;u++) {
        //to select a dest
        for(int v=0; v < matrix.length;v++) {
            if(u == v || u == k || v == k || matrix[u][k] == -1 || matrix[k][v] == -1) {
                continue;
            }
            else if(matrix[u][v] == -1 || matrix[u][k] + matrix[k][v] < matrix[u][v]) {
                matrix[u][v] = matrix[u][k] + matrix[k][v];
            }
        }
    }
}
```

$k = 0$

	0	1	2	3
0	0	3	-1	7
1	8	0	2	15
2	5	8	0	1
3	2	5	-1	0

Alien dictionary

wrt, wrf, er, ett, rftt

5
wrt wrf er ett rftt

t < f

r < t

r < t < f

w < e < r

(i) graph

(ii) topological sort

w < e < r < t < f

wrt, wrF, er, ett, rjtt

Hashmap <Char, Arraylist <char>> graph;

wrt wr

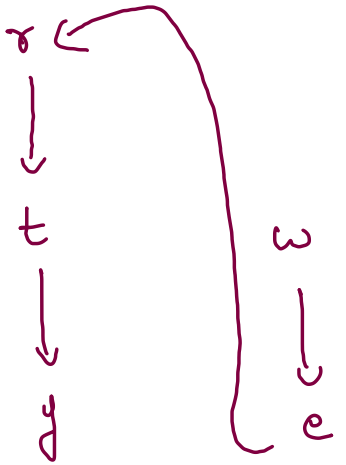
w →

r →

t →

j →

e →



wertf