Lab11 Solution

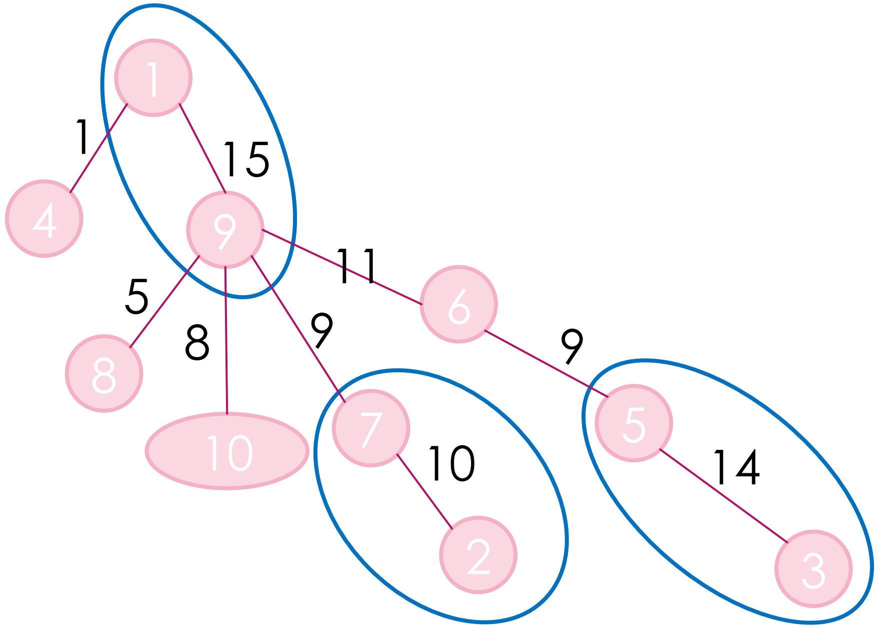
YAO ZHAO

Lab11.A: Match on Tree

- Yan got a tree as his birthday gift.
- The tree has N nodes and $N-1$ undirected weighted edges.
- Yan decided to find some matches on the tree. A match consists of a pair of nodes (u, v), such that there exists some edge connecting node u and node v. The value of this match is defined as the weight of that edge.
- Yan can make several matches, as long as **each node belongs to no more than 1 match**. Let S be the sum of values of all matches he makes. Help Yan find the maximum of S.

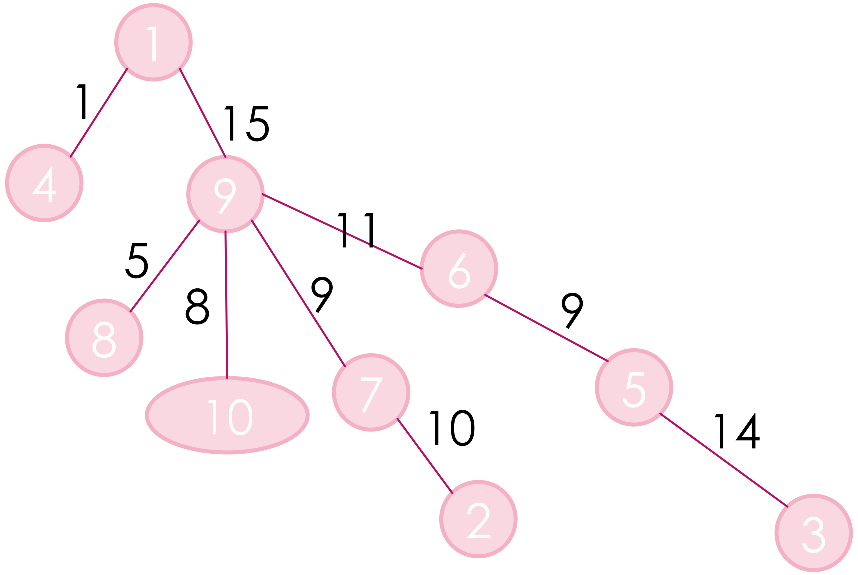
Sample Input

10
9 6 11
9 1 15
9 7 9
9 10 8
7 2 10
1 4 1
9 8 5
6 5 9
5 3 14



Sample Output

39

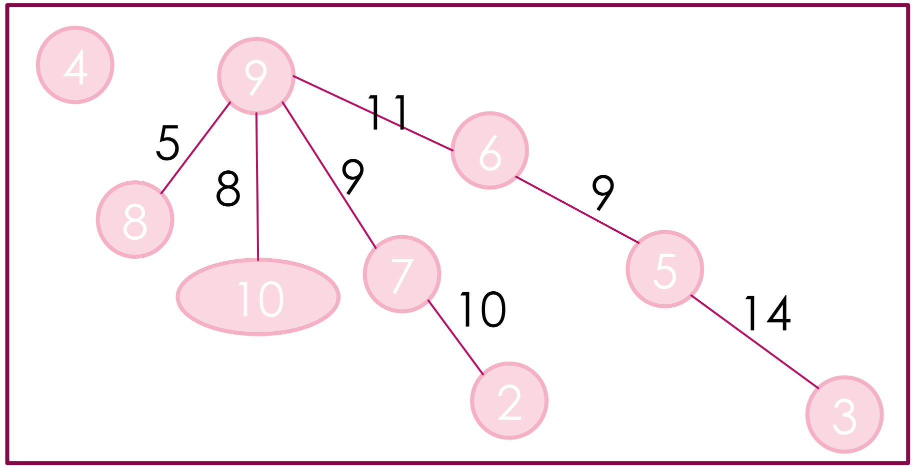


let node 1 to be the root of the tree

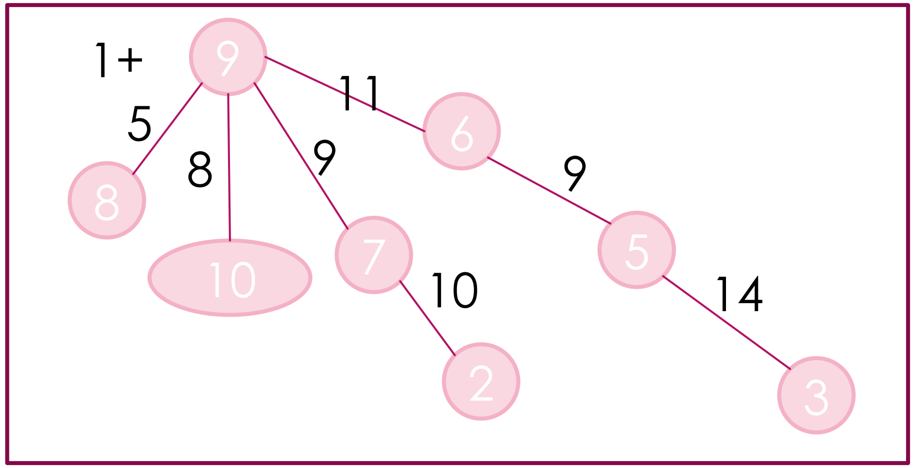
for node 1, there are 3 choices: not pick node 1, choose node1 and node 4, choose node 1 and node 9

The remaining subtrees:

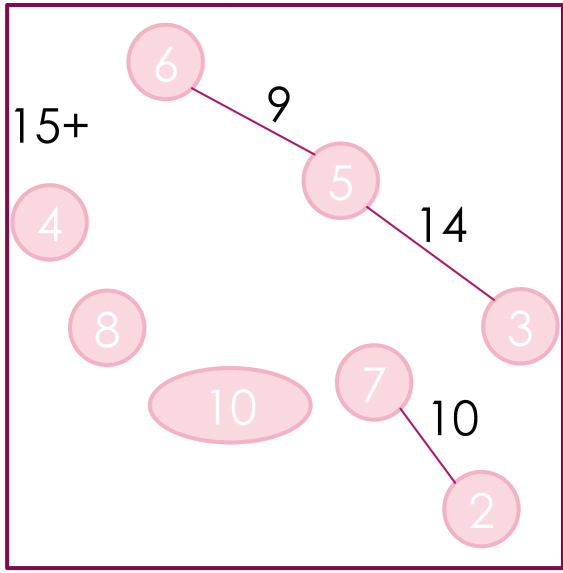
not pick node 1

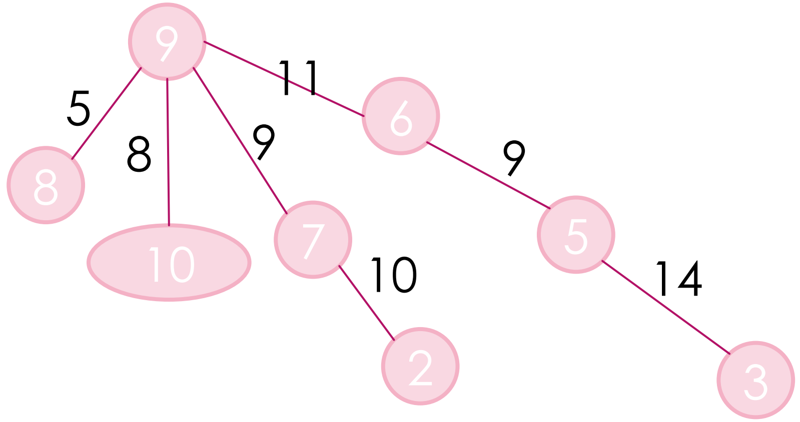


match 1 4

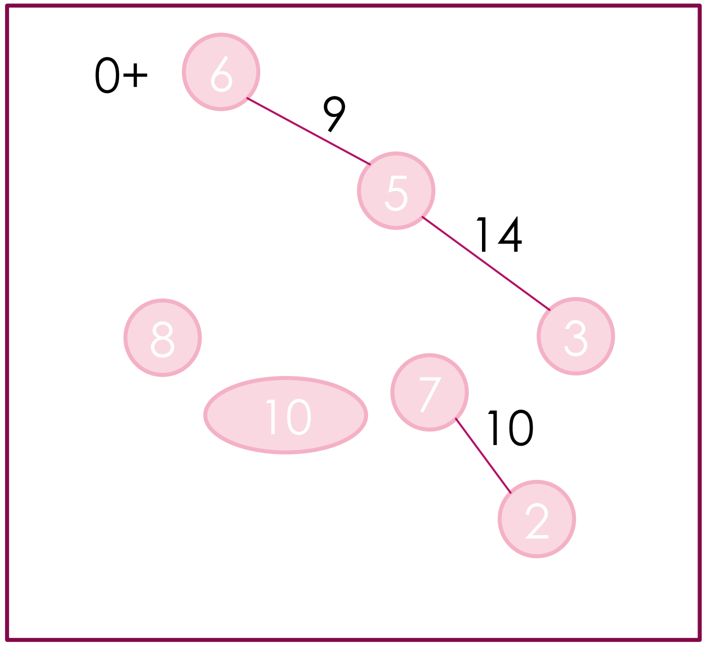


match 1 9

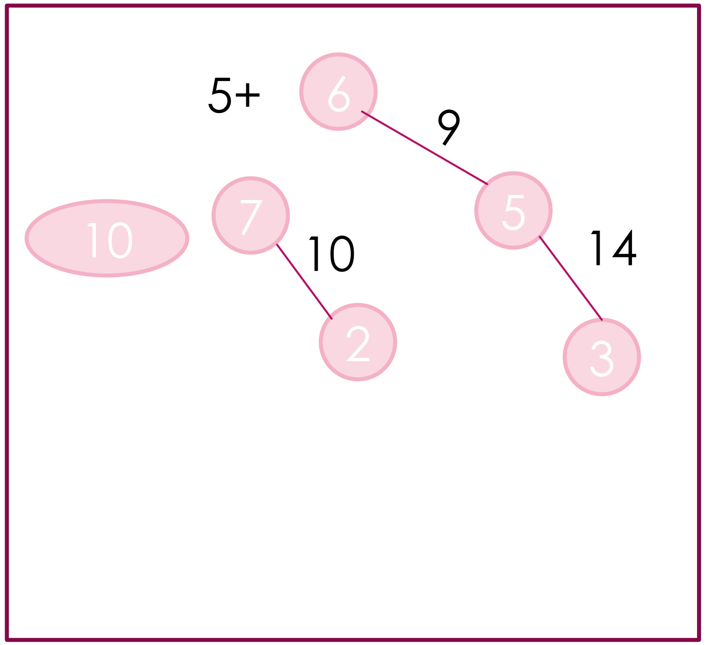




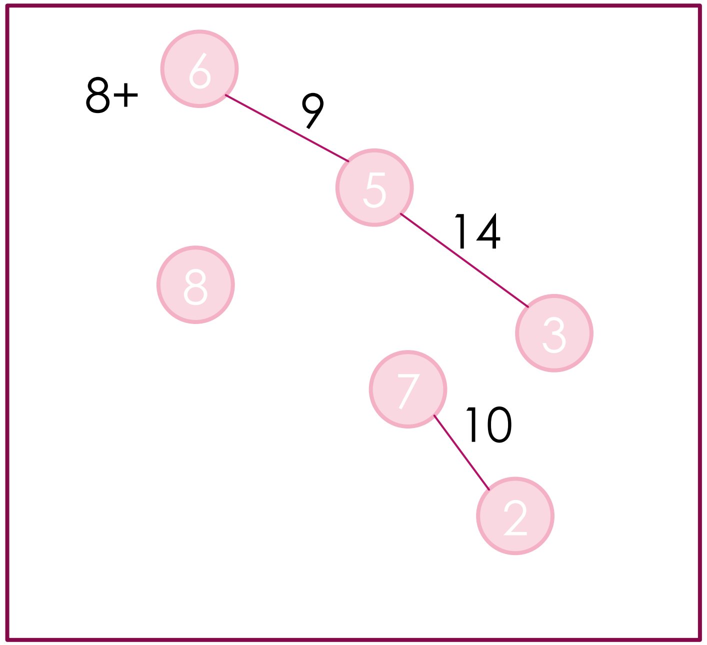
not pick node 9



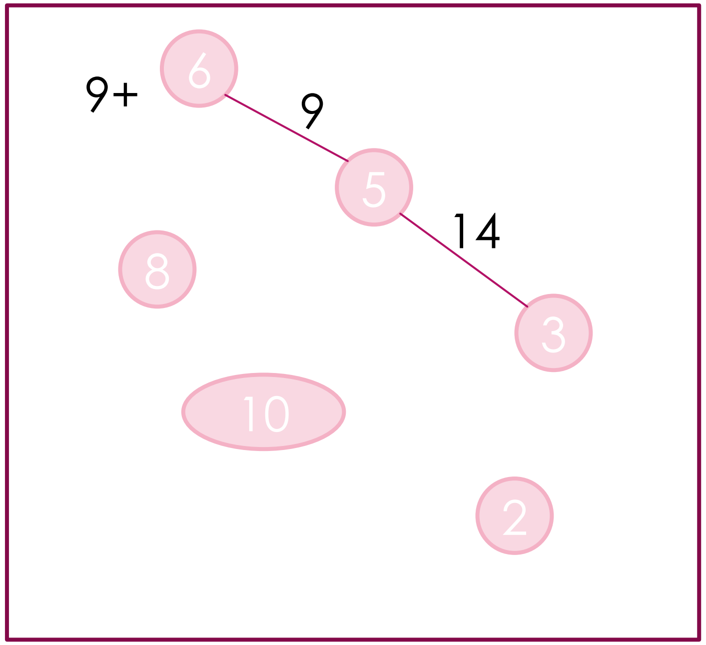
match 9 8



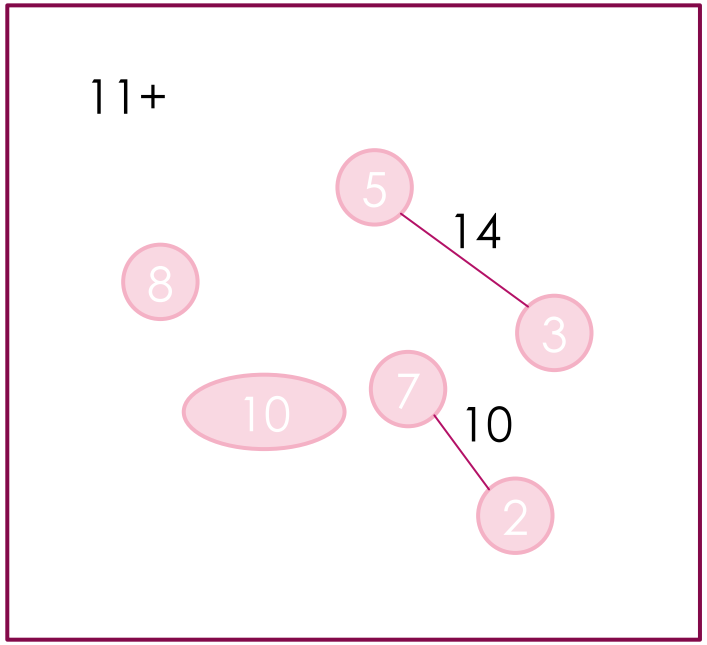
match 9 10



match 9 7



match 9 6



let $opt[i]$ present the maximum of S of the tree which root node is node i

$$
opt[i][1] = \begin{cases} 0 \quad if\ i\ is\ a\ leaf\ node \\ \text{let } opt[i][0] = \sum_{j=1}^{k} opt\left[v_j\right][1] \\ \max\{opt[i][0],\ w_{i,v_1} + opt[i][0] - opt[v_1][1] + opt[v_1][0],\ \ldots,\ w_{i,v_k} + opt[i][0] - opt[v_k][1] + opt[v_k][0]\} \end{cases}
$$

$assume\ k\ child\ nodes\ v_1, v_2, \ldots, v_k\ connect\ node\ i$

Lab11.B: Strange Courses

- ZT's college has N distinctive courses and M dependencies. Each dependency is described as (u, v), which means that a student must learn course v before learning course u.
- Strangely, those dependencies may form cycles, which is not reasonable for a modern college.
- Therefore, ZT plans to remove none, some, or all of those M dependencies. A removal is **good** if no cycle exists in the remaining dependencies.
- For a **good** removal, its **flexibility** is defined as the number of permutations of $1 \ldots N$, such that a student can learn the N courses following the order of permutation without violating the remaining dependencies.
- ZT wishes to know the sum of **flexibility** of all **good** removals, modulo $10^9 + 7$.

Sample 1 Input
2 2
1 2
2 1

remove 0 : have circle, not a good removal

remove 1 → 2: **flexibility=1** permutation:21

remove 2 → 1: **flexibility=1** permutation:12

remove 1 → 2 and 2→1: **flexibility=2** permutation: 12 and 21

the sum of flexibility of all good removals: **4**

Sample 1 Output
4

1 start from node 1, no edge here

then try to add node 2, from node 1, how many ways let 1 before 2?

There is 1 edge from 1 to 2, there is **2** way let 1 before 2:
remove the edge 1→2, 1 way
keep the edge, 1 way

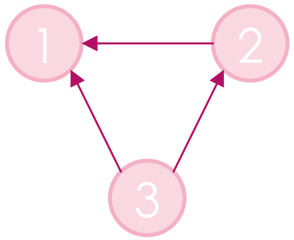
2 start from node 2, no edge here

then try to add node 1, from node 2, how many ways let 2 before 1?

There is 1 edge from 2 to 1, there is **2** way let 2 before 1:
remove the edge 2→1, 1 way
keep the edge, 1 way

4

Sample 3 Input
3 3
1 2
2 3
1 3



initial: Assume there are no nodes in the initial state

{}
only 1 way

add a node to the state

add a node 1

{1}
1 way

add 2

{1}→2
there is no edge from 1 to 2, only {1} way make 1 before 2
1 way

add 3

{1}→3
there is no edge from 1 to 3, only {1} way make 1 before 3
1 way

add a node 2

{2}
1 way

add 1

{2}→1
there is 1 edge from 1 to 2, {2}*2 way make 2 before 1, keep the edge or remove the edge
1*2 way

add 3

{2}→3
there is no edge from 1 to 3, only {2} way make 1 before 3
1 way

add a node 3

{3}
1 way

add 1

{3}→1
there is 1 edge from 3 to 1, {3}*2 way make 3 before 1, keep the edge or remove the edge
1*2 way

add 2

{3}→2
there is 1 edge from 3 to 2, {3}*2 way make 3 before 2, keep the edge or remove the edge
1*2 way

{1}→2
there is no edge from 1 to 2, only {1} way make 1 before 2
1 way

{1}→3
there is no edge from 1 to 3, only {1} way make 1 before 3
1 way

{2}→1
there is 1 edge from 1 to 2, {1}*2 way make 2 before 1, keep the edge or remove the edge
1*2 way

{2}→3
there is no edge from 1 to 3, only {1} way make 1 before 3
1 way

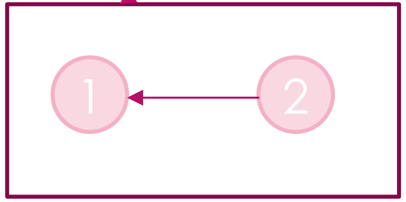
{3}→1
there is 1 edge from 3 to 1, {1}*2 way make 3 before 1, keep the edge or remove the edge
1*2 way

{3}→2
there is 1 edge from 3 to 2, {1}*2 way make 3 before 2, keep the edge or remove the edge
1*2 way

+

3 ways

answer of the following sub graph

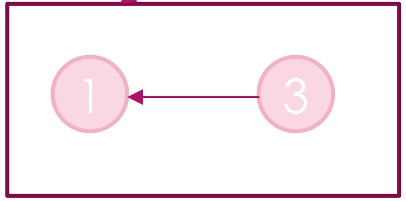


{1,2}

+

3 ways

answer of the following sub graph

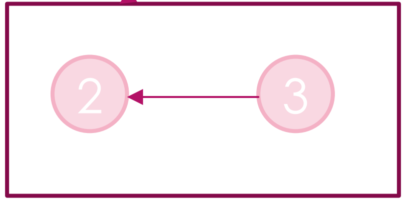


{1,3}

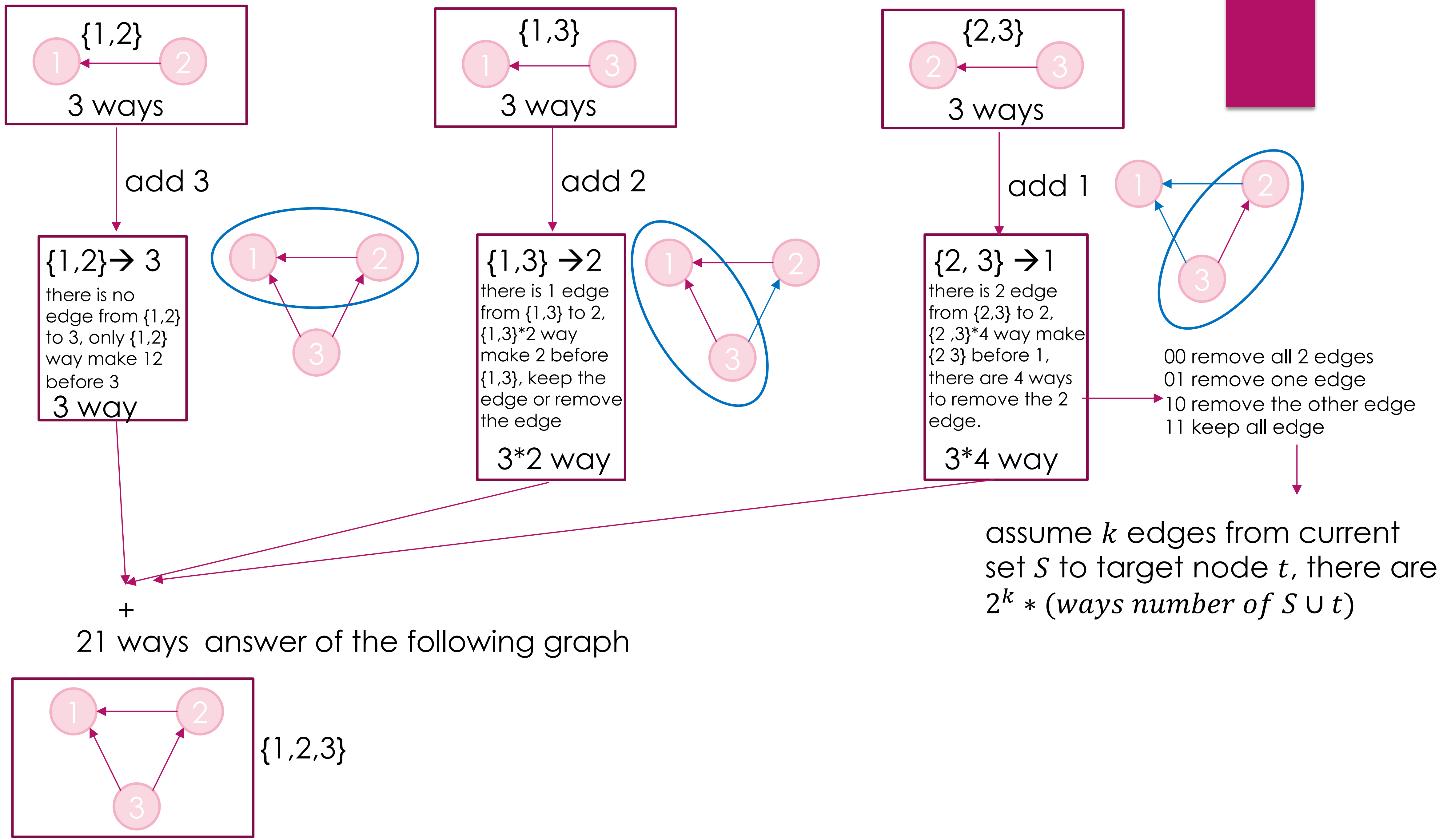
+

3 ways

answer of the following sub graph



{2,3}


{1,2}
1
2
3 ways
add 3
{1,2}→ 3
there is no edge from {1,2} to 3, only {1,2} way make 12 before 3
3 way
1
2
3
{1,3}
1
3
3 ways
add 2
{1,3} →2
there is 1 edge from {1,3} to 2, {1,3}*2 way make 2 before {1,3}, keep the edge or remove the edge
3*2 way
1
2
3
{2,3}
2
3
3 ways
add 1
{2, 3} →1
there is 2 edge from {2,3} to 2, {2 ,3}*4 way make {2 3} before 1, there are 4 ways to remove the 2 edge.
3*4 way
1
2
3
00 remove all 2 edges
01 remove one edge
10 remove the other edge
11 keep all edge
assume k edges from current set S to target node t, there are $2^k * (ways\ number\ of\ S \cup t)$
+
21 ways answer of the following graph
1
2
3
{1,2,3}

How to present the states? There may be many many different states:

{}{1}{2}{3}{1,2}{1,3}{2,3}{1,2,3}….

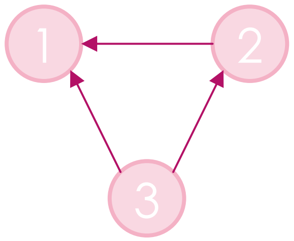
n nodes → 2^n $states$

bits:

0	0	0	{}
0	0	1	{1}
0	1	0	{2}
1	0	0	{3}
0	1	1	{1,2}
1	0	1	{1,3}
1	1	0	{2,3}
1	1	1	{1,2,3}

Sample 3 Input

3 3
1 2
2 3
1 3



	1	2	3
1		1	1
2			1
3			

	dp[]:	0(000)	1(001)	2(010)	3(011)	4(100)	5(101)	6(110)	7(111)
	initial: {}	1	0	0	0	0	0	0	0
iter 1:	$0(000)\ \{\}+\left\{\begin{matrix}1\\2\\3\end{matrix}\right.$	1	**1**	**1**	0	**1**	0	0	0
iter 2:	$1(001)\ \{1\}+\left\{\begin{matrix}2\\3\end{matrix}\right.$	1	1	1	**1**	1	**1**	0	0
iter 3:	$2(010)\ \{2\}+\left\{\begin{matrix}1\\3\end{matrix}\right.$	1	1	1	**3**	1	1	**1**	0
iter 4:	3(011){1,2}+3	1	1	1	3	1	1	1	**3**
iter 5:	$4(100)\{3\}+\left\{\begin{matrix}1\\2\end{matrix}\right.$	1	1	1	3	1	**3**	**3**	3
iter 6:	5(101){1,3}+2	1	1	1	3	1	3	3	**9**
iter 7:	6(110){2,3}+1	1	1	1	3	1	3	3	**21**