Lab3 Solution&Hint

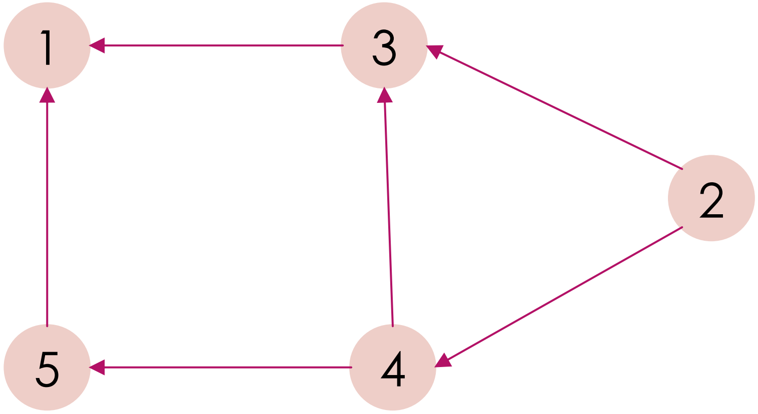
YAO ZHAO

Lab3.A: Biology and CS

- There are M predation relationships among N species in some fauna. It is guaranteed that there is no cyclic predation in the food web.
- A food chain is defined by a sequence of species $[a_1, a_2, \dots, a_k]$, where
  - nothing can prey on a_1
  - a_k can prey on nothing
  - a_i can prey on a_{i+1} for $i = 1, 2, \dots, k-1$
- Two food chains are different, if and only if their sequences are different.
- Now each of these N species wants to know how many food chains involve with it, module $10^9 + 7$.

Sample Input 1

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

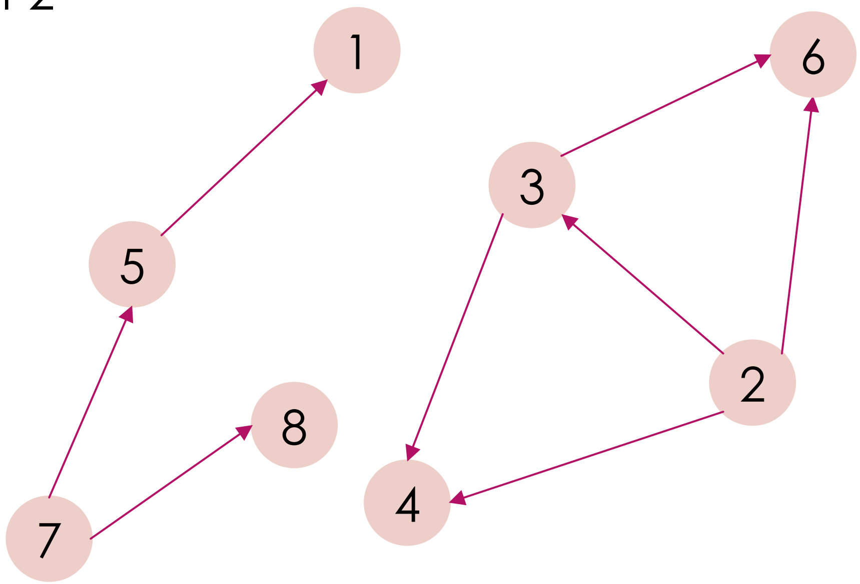


species	food chains	number
1	[2,3,1] [2,4,3,1] [2,4,5,1]	3
2	[2,3,1] [2,4,3,1] [2,4,5,1]	3
3	[2,3,1] [2,4,3,1]	2
4	[2,4,3,1] [2,4,5,1]	2
5	[2,4,5,1]	1

Sample Output 1
3 3 2 2 1

Sample Input 2

8 8
2 4
7 5
3 4
3 6
2 6
7 8
2 3
5 1



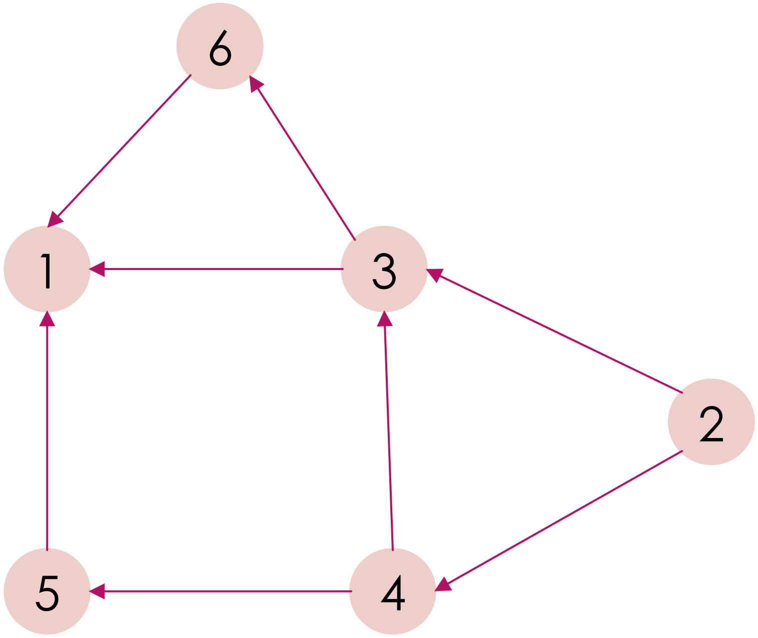
species	food chains	number
1	[7,5,1]	1
2	[2,4] [2,3,4] [2,6] [2,3,6]	4
3	[2,3,4] [2,3,6]	2
4	[2,4] [2,3,4]	2
5	[7,5,1]	1
6	[2,6] [2,3,6]	2
7	[7,5,1] [7,8]	2
8	[7,8]	1

Sample Output 2

1 4 2 2 1 2 2 1

Sample Input 3

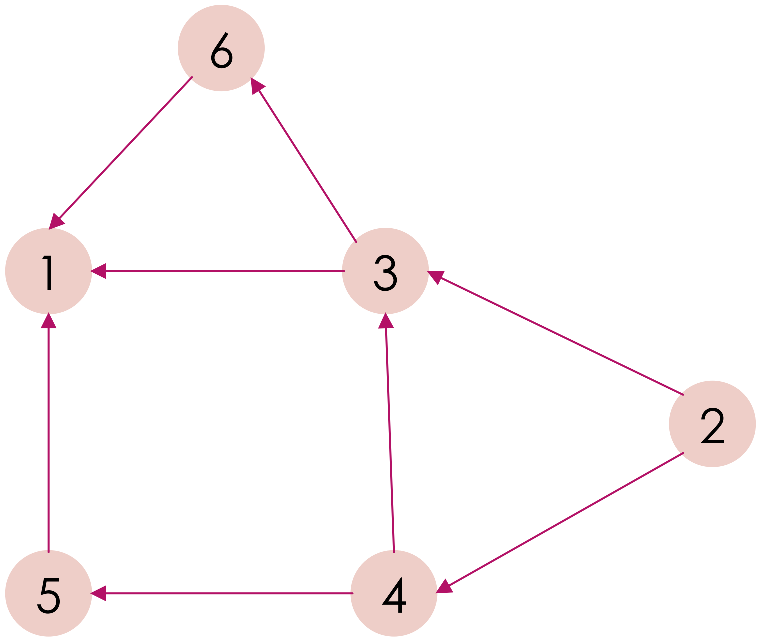
6 8
4 5
4 3
2 4
3 1
2 3
5 1
3 6
6 1



species	food chains	number
1	[2,3,1] [2,4,3,1] [2,4,5,1] [2,3,6,1] [2,4,3,6,1]	5
2	[2,3,1] [2,4,3,1] [2,4,5,1] [2,3,6,1] [2,4,3,6,1]	5
3	[2,3,1] [2,4,3,1] [2,3,6,1] [2,4,3,6,1]	4
4	[2,4,3,1] [2,4,5,1] [2,4,3,6,1]	3
5	[2,4,5,1]	1
6	[2,3,6,1] [2,4,3,6,1]	2

Try to find all topo order and update the chains number of each node

Find all nodes with no incoming edges
Init their chains number to 1 and put them to queue Q
While Q is not empty:
    get a node e from Q
    for each child c of node:
        c.indegree ← c.indegree -1
        c.chainsnumber← c.chainsnumber+e. chainsnumber
        if c.indegree == 0
            put c to Q



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

Why? ← **Can't find all chains number** ←

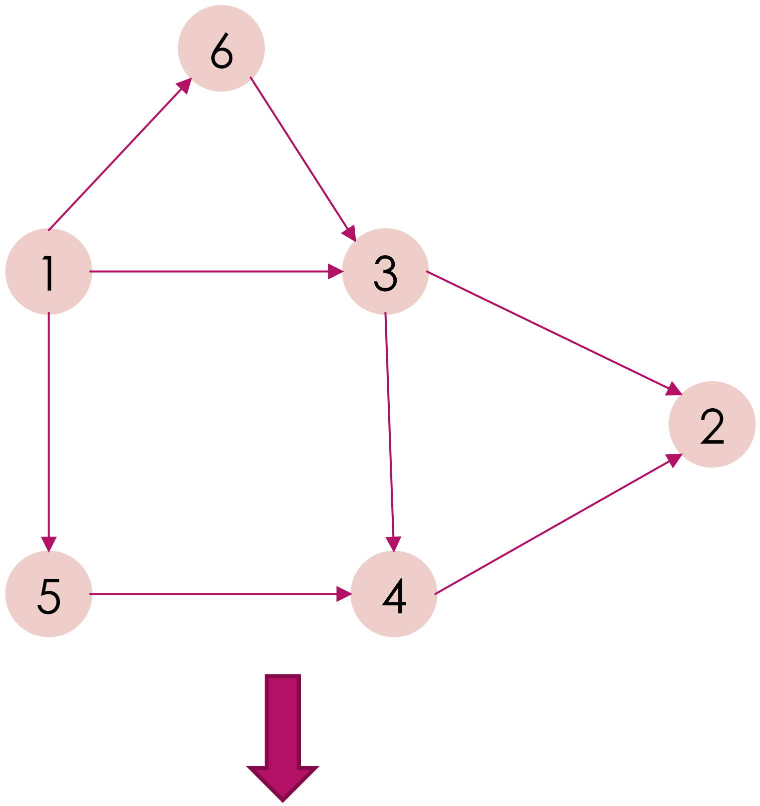
species	food chains	number	result
1	[2,3,1] [2,4,3,1] [2,4,5,1] [2,3,6,1] [2,4,3,6,1]	5	**5**
2	[2,3,1] [2,4,3,1] [2,4,5,1] [2,3,6,1] [2,4,3,6,1]	5	**1**
3	[2,3,1] [2,4,3,1] [2,3,6,1] [2,4,3,6,1]	4	**2**
4	[2,4,3,1] [2,4,5,1] [2,4,3,6,1]	3	**1**
5	[2,4,5,1]	1	**1**
6	[2,3,6,1] [2,4,3,6,1]	2	**2**

Only find the combination before the node

How to find all?

Reverse the edges and execute again:

Find all nodes with no incoming edges
Init their chains number to 1 and put them to queue Q
While Q is not empty:
    get a node e from Q
    for each child c of node:
        c.indegree ← c.indegree -1
        c.chainsnumber← c.chainsnumber+e. chainsnumber
        if c.indegree == 0
            put c to Q

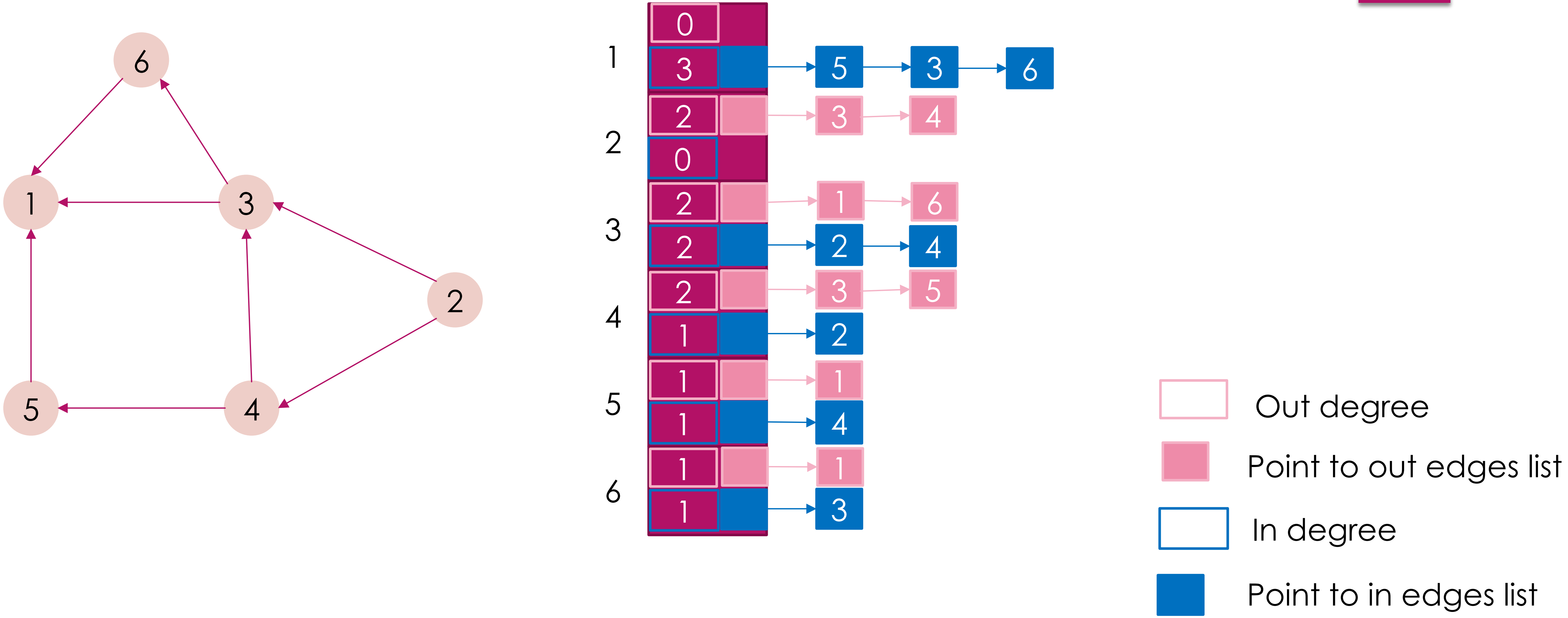


1	2	3	4	5	6
1	5	2	3	1	1

species	food chains	number	Result	Reverse-result
1	[2,3,1] [2,4,3,1] [2,4,5,1] [2,3,6,1] [2,4,3,6,1]	5	**5**	**1**
2	[2,3,1] [2,4,3,1] [2,4,5,1] [2,3,6,1] [2,4,3,6,1]	5	**1**	**5**
3	[2,3,1] [2,4,3,1] [2,3,6,1] [2,4,3,6,1]	4	**2**	**2**
4	[2,4,3,1] [2,4,5,1] [2,4,3,6,1]	3	**1**	**3**
5	[2,4,5,1]	1	**1**	**1**
6	[2,3,6,1] [2,4,3,6,1]	2	**2**	**1**

number = result * Reverse-result

Recommended data structure

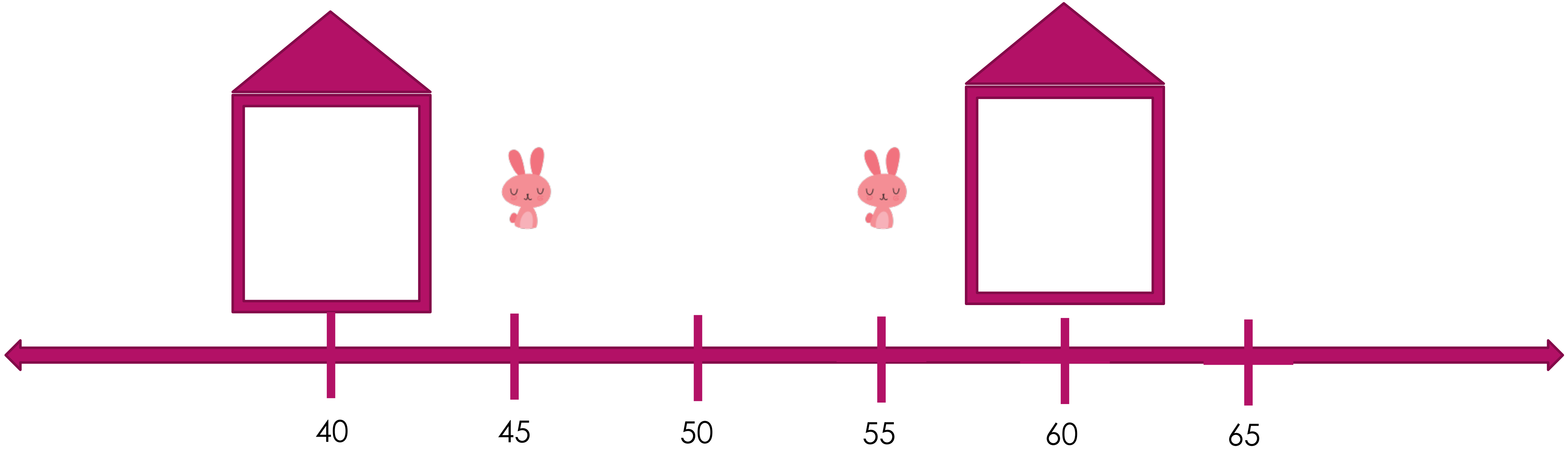


Lab3.B: Legendary Grabbing Machine

- Satori is a hunter who likes hunting bunnies.
- Initially at moment 0, there are N bunnies, the i^{th} of which is at position p_i.
- There are also M nests, the i^{th} of which is at position q_i. Each nest can hold at most C bunnies.
- Each bunny can move at most 1 unit of distance within 1 unit of time. Once a bunny enters some nest, it will be completely safe from Satori.
- Satori's **Legendary Grabbing Machine** takes T units of time to charge. Once the machine finishes charging at moment T, all bunnies that are out of nest will be captured. Note that bunnies entering nest at moment T will be safe.
- The bunnies very are united. They wish to know the maximal number of safe bunnies if they move optimally.

Input:
2 2 1 5
45 55
40 60

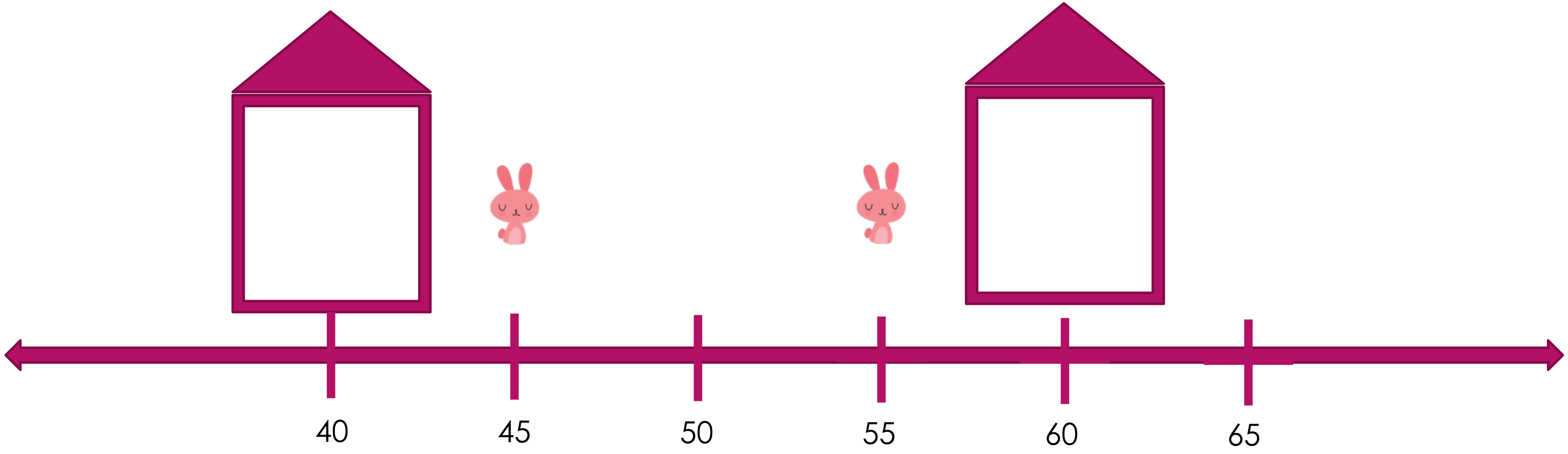
Legendary Grabbing Machine



Input:
2 2 1 5
45 55
40 60

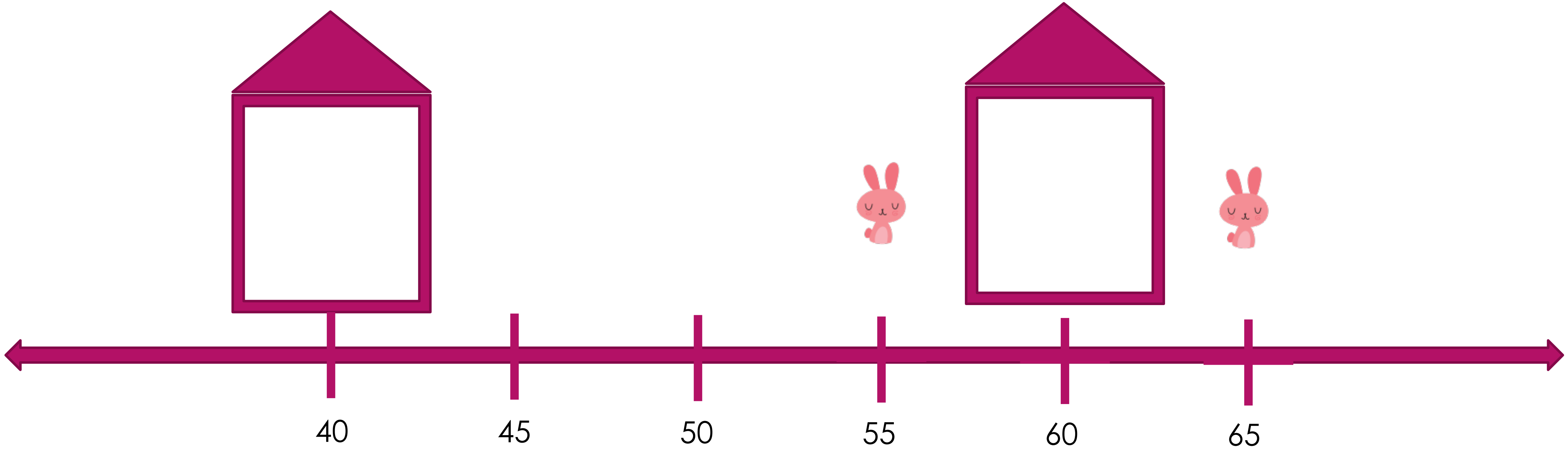
Output:
2

Legendary Grabbing Machine



Input:
2 2 1 5
55 65
40 60

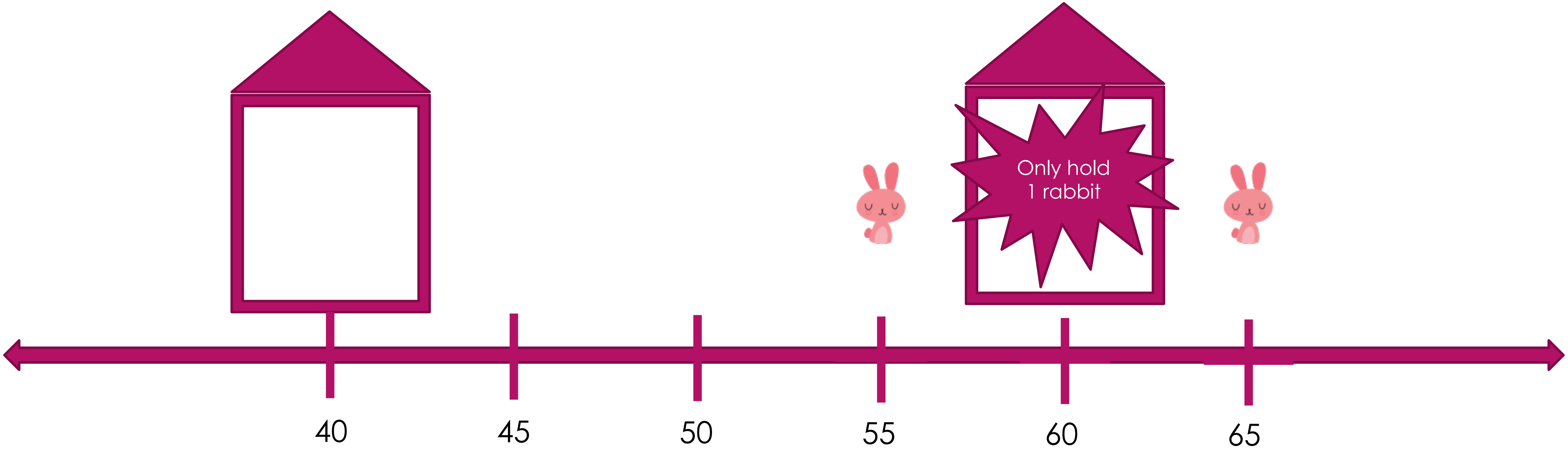
Legendary Grabbing Machine



Legendary Grabbing Machine

Input:
2 2 1 5
55 65
40 60

Output:
1



Hint:

Sort the nests by position
Sort the bunnies by position
Start from one direction or another, greedily fill the nests