CCC JUNIOR

Graph

今日课程预览

- Homework from Week 4
- Graph Structure
- Graph 2D Array
- Graph ArrayList
- Graph with Weights
- Distance in Graph – Flyod Warshall Algorithm

CCC 2015 S1 Zero That Out

https://www.cemc.uwaterloo.ca/contests/computing/2015/stage%201/seniorEn.pdf

- **Problem Description**

Your boss has asked you to add up a sequence of positive numbers to determine how much money your company made last year. Unfortunately, your boss reads out numbers incorrectly from time to time. Fortunately, your boss realizes when an incorrect number is read and says “zero”, meaning “ignore the current last number.” Unfortunately, your boss can make repeated mistakes, and says “zero” for each mistake. For example, your boss may say “One, three, five, four, zero, zero, seven, zero, zero, six”, which means the total is 7 as explained in the following chart:

Boss statement(s)	Current numbers	Explanation
“One, three, five, four”	1, 3, 5, 4	Record the first four numbers.
“zero, zero”	1, 3	Ignore the last two numbers.
“seven”	1, 3, 7	Record the number 7 at the end of our list.
“zero, zero”	1	Ignore the last two numbers.
“six”	1, 6	We have read all numbers, and the total is 7.

Input and Output Specification

- At any point, your boss will have said at least as many positive numbers as “zero” statements. If all positive numbers have been ignored, the sum is zero. Write a program that reads the sequence of boss statements and computes the correct sum.
- Input Specification

The first line of input contains the integer K (1 ≤ K ≤ 100 000) which is the number of integers (including “zero”) your boss will say. On each of the next K lines, there will either be one integer between 1 and 100 (inclusive), or the integer 0.

- Output Specification

The output is one line, containing the integer which is the correct sum of the integers read, taking the “zero” statements into consideration. You can assume that the output will be an integer in the range 0 and 1 000 000 (inclusive).

Sample Input/Output

- Sample Input 1

  4

  3

  0

  4

- Output for Sample Input 1

  0

- Sample Input 2

  10

  1

  3

  5

  4

  0

  0

  7

  0

  0

  6

- Output for Sample Input 2

  7

CCC 2005 J3 Returning Home

https://cemc.math.uwaterloo.ca/contests/computing/2005/stage1/juniorEn.pdf

- **Problem Description**

Jane's family has just moved to a new city and today is her first day of school. She has a list of instructions for walking from her home to the school. Each instruction describes a turn she must make. For example, the list

```
R
QUEEN
R
FOURTH
R
SCHOOL
```

means that she must turn right onto Queen Street, then turn right onto Fourth Street, then finally turn right into the school. Your task is to write a computer program which will create instructions for walking in the opposite direction: from her school to her home. The input and output for your program will be formatted like the samples below. You may assume that Jane's list contains at least two but at most five instructions, and you may assume that each line contains at most 10 characters, all of them capital letters. The last instruction will always be a turn into the "SCHOOL".

Sample Input and Output

- **Sample Input 1**

```
R
QUEEN
R
FOURTH
R
SCHOOL
```

- **Sample Output for Sample Input 1**

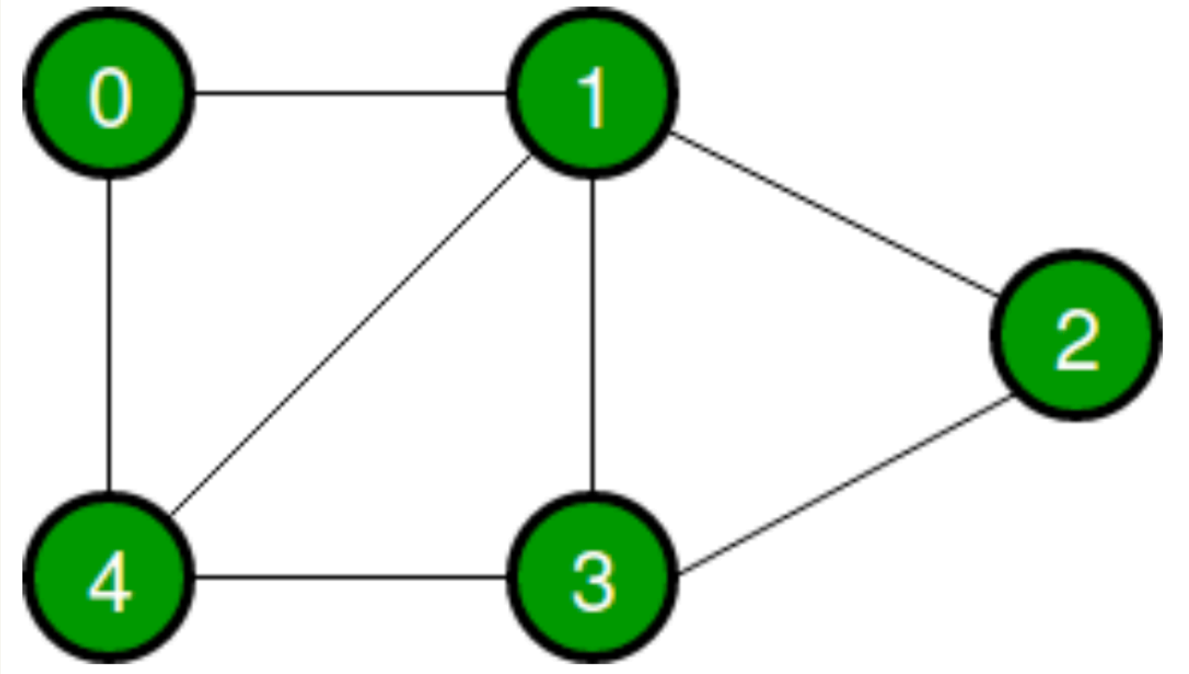
```
Turn LEFT onto FOURTH street.
Turn LEFT onto QUEEN street.
Turn LEFT into your HOME.
```

- **Sample Input 2**

```
L
MAIN
R
SCHOOL
```

- **Sample Output for Sample Input 2**

```
Turn LEFT onto MAIN street.
Turn RIGHT into your HOME.
```



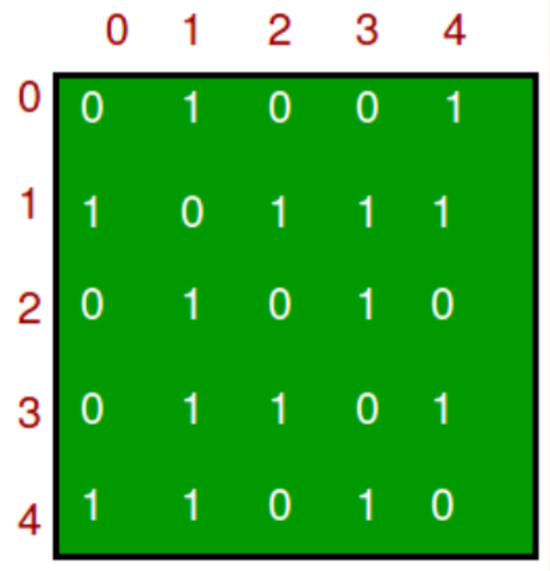
GRAPH

2D Array/ArrayList

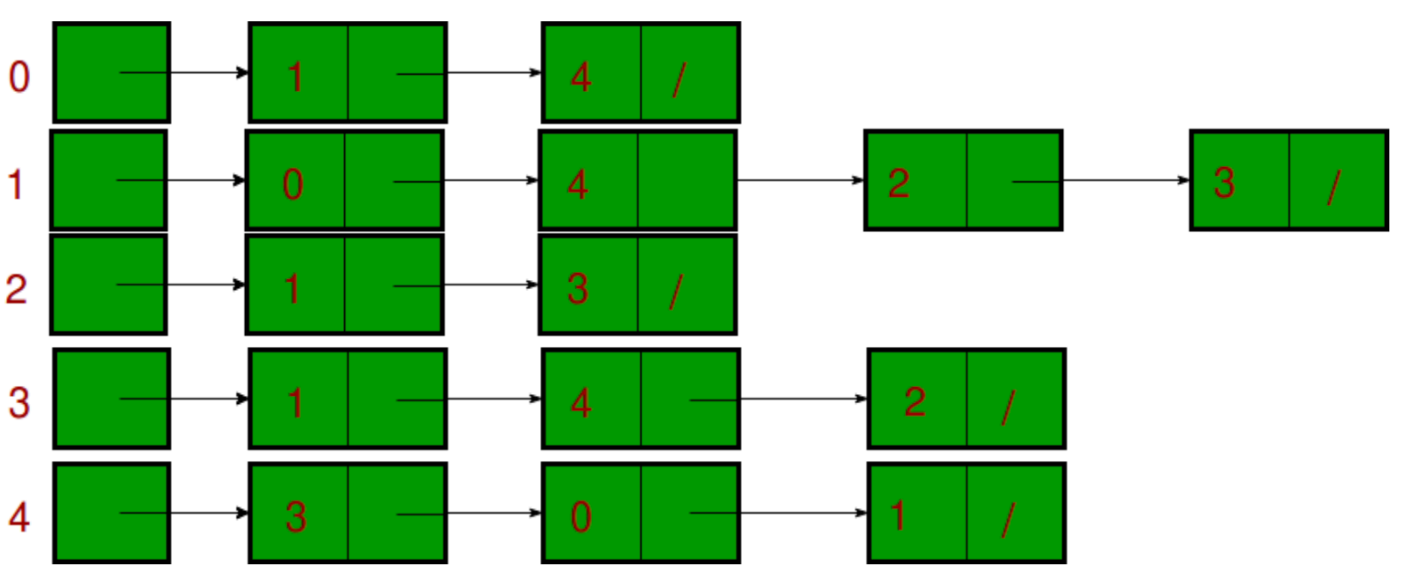
Code Representing Graph

- 2D Array
- *int[] graph = new int[N]*

- ArrayList of ArrayList
- *ArrayList<ArrayList> graph = new ArrayList<>()*



	0	1	2	3	4
0	0	1	0	0	1
1	1	0	1	1	1
2	0	1	0	1	0
3	0	1	1	0	1
4	1	1	0	1	0



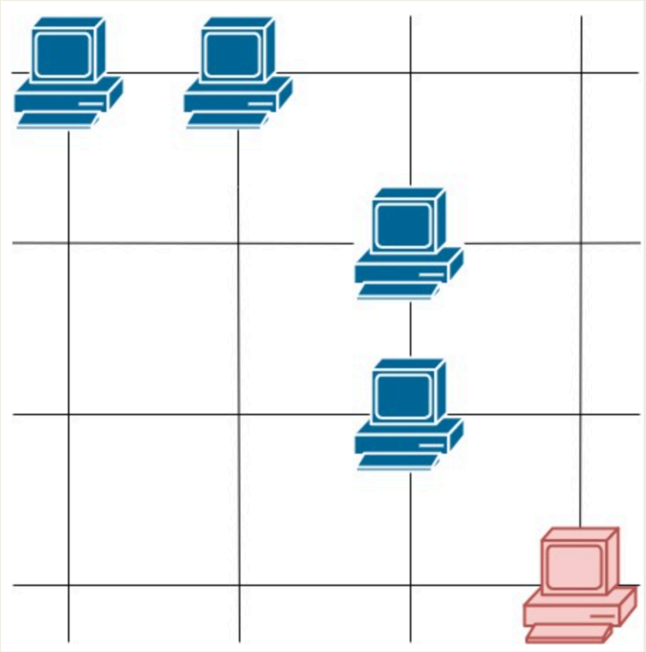
Graph 2D Array Example

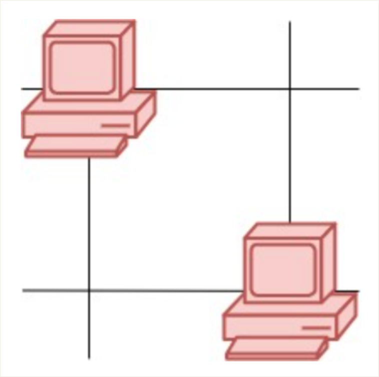
- You are given a map of a server center, represented as a m * n integer matrix grid, where 1 means that on that cell there is a server and 0 means that it is no server. Two servers are said to communicate if they are on the same row or on the same column. Return the number of servers that communicate with any other server.

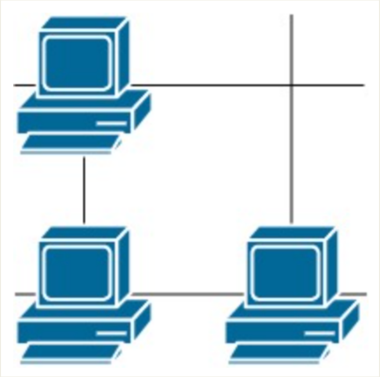
  Input: grid = [[1,0],[0,1]] **Output:** 0

  Input: grid = [[1,0],[1,1]] **Output:** 3

  Input: grid = [[1,1,0,0],[0,0,1,0],[0,0,1,0],[0,0,0,1]] **Output:** 4







CCC 2009 J5 Degree of Separation

https://www.cemc.uwaterloo.ca/contests/computing/2009/stage1/juniorEn.pdf

- **Problem Description**

The main socializing tool for students today is Facebook. There are many interesting computational questions connected to Facebook, such as the “degree of separation” between two people. For example, in the diagram below, there are many different paths between Abby and Alberto.

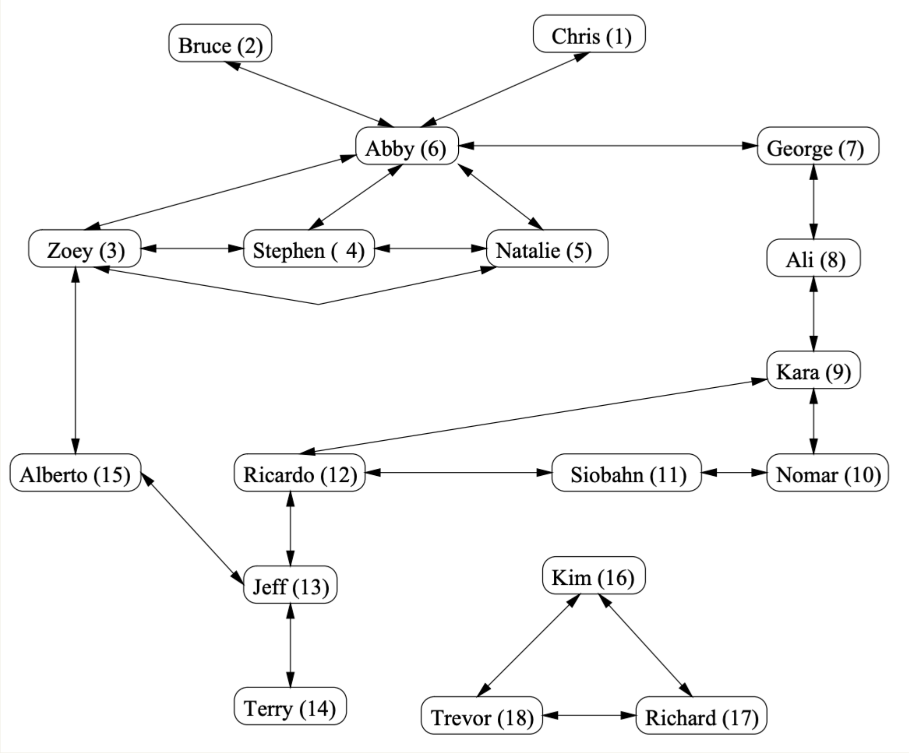
Some of these paths are:

- Abby → Zoey → Alberto
- Abby → Natalie → Zoey → Alberto
- Abby → George → Ali → Kara → Richardo → Jeff → Alberto

The shortest path between Abby and Alberto has two steps (Abby → Zoey, and Zoey → Alberto), so we say the degree of separation is 2. Additionally, Alberto would be a friend of a friend of Abby.

CCC 2009 J5 Degree of Separation

https://www.cemc.uwaterloo.ca/contests/computing/2009/stage1/juniorEn.pdf



You can assume an initial configuration of who is friends with who as outlined in the diagram above. You will need to store these relationships in your program. These relationships can change though, and your program needs to handle these changes.

In particular, friendships can begin, possibly with new people. Friendships can end. You should be able to find friends of friends and determine the degree of separation between two people.

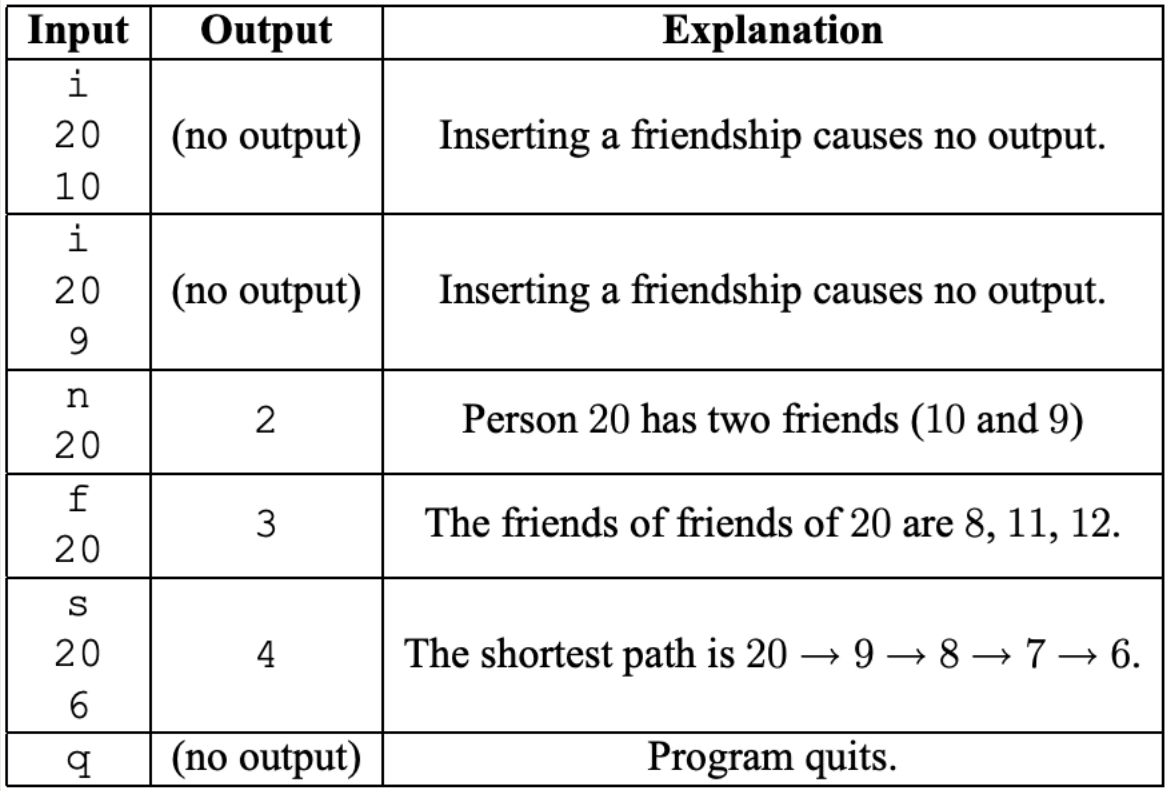
Input/Output Specification

- **Input/Output Specification**

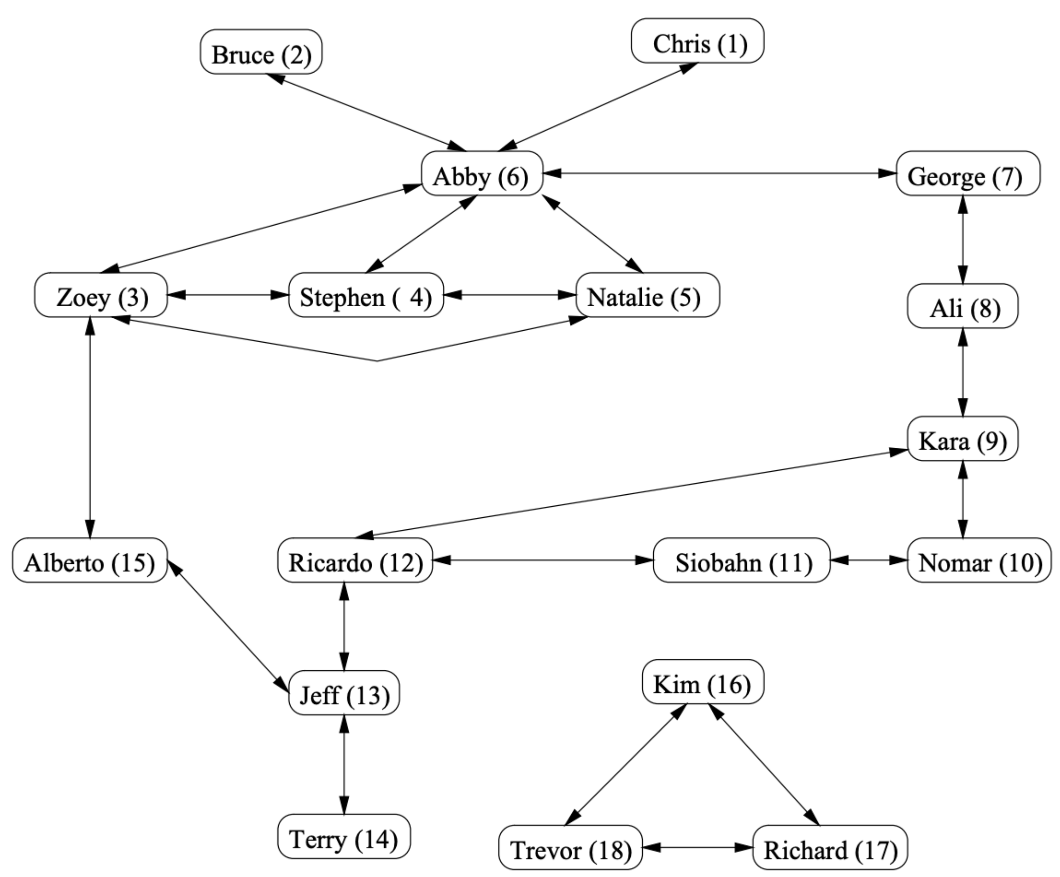
Your program will read in six possible commands, with the action to be performed by your program outlined below. You may assume that x and y are integers, with x 6= y, x ≥ 1, y ≥ 1, x < 50 and y < 50. You may also assume that instructions (i, d, n, f, s, q) occur one per line and parameters (zero, one or two integers) occur one per line.

- i x y – make person x and person y friends. If they are already friends, no change needs to be made. If either x or y is a new person, add them.
- d x y – delete the friendship between person x and person y.
- n x – output the number of friends that person x has.
- f x – output the number of “friends of friends” that person x has. Notice that x and direct friends of x are not counted as “friends of friends.”
- s x y – output the degree of separation between x and y. If there is no path from x to y, output Not connected.
- q – quit the program.

Sample Input/Output

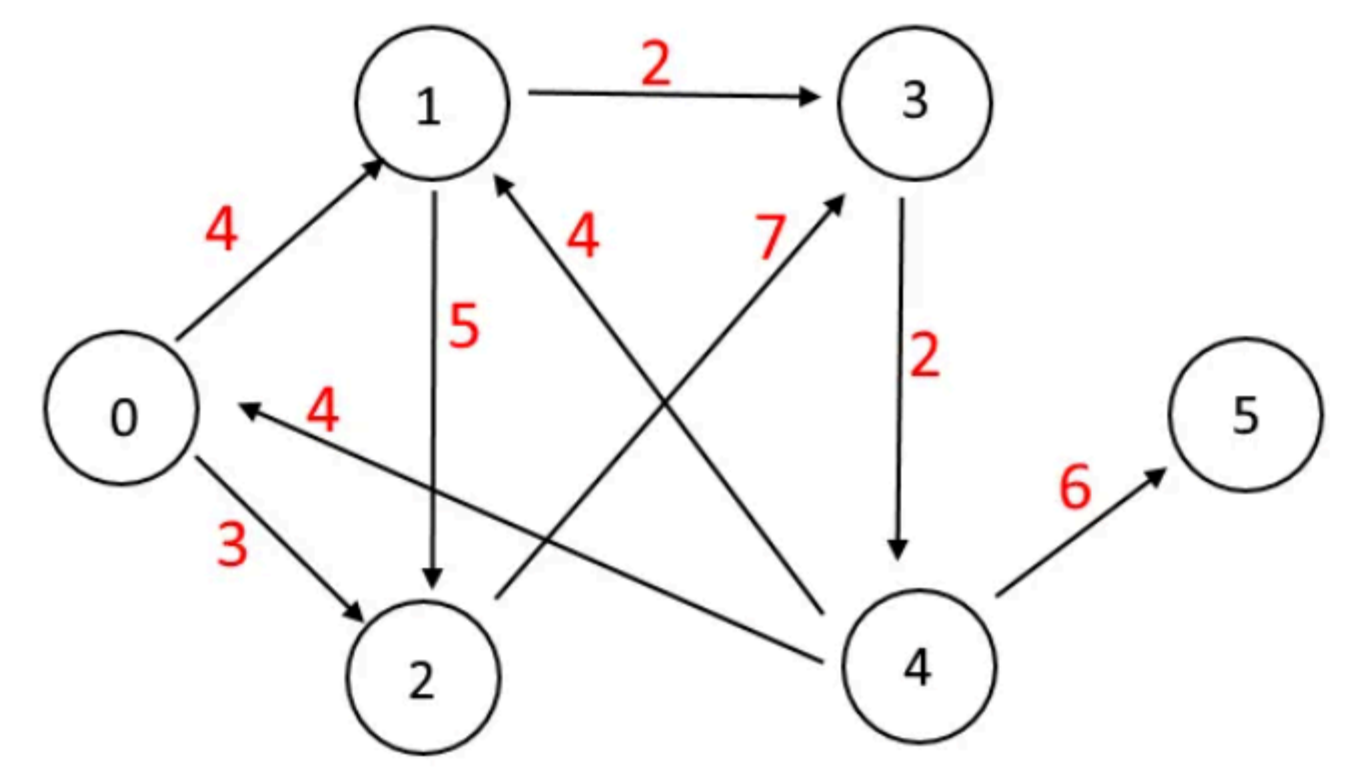


Input	Output	Explanation
i 20 10	(no output)	Inserting a friendship causes no output.
i 20 9	(no output)	Inserting a friendship causes no output.
n 20	2	Person 20 has two friends (10 and 9)
f 20	3	The friends of friends of 20 are 8, 11, 12.
s 20 6	4	The shortest path is 20 → 9 → 8 → 7 → 6.
q	(no output)	Program quits.



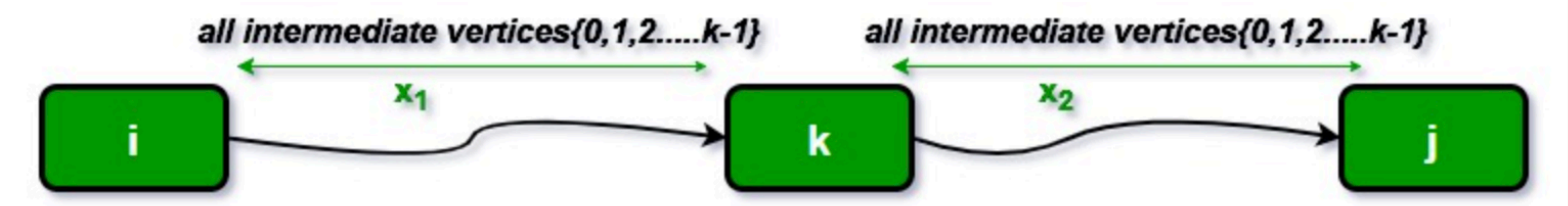
Some Graph with Weights

- ArrayList<int[]> graph = new ArrayList<>();



Floyd–Warshall Algorithm

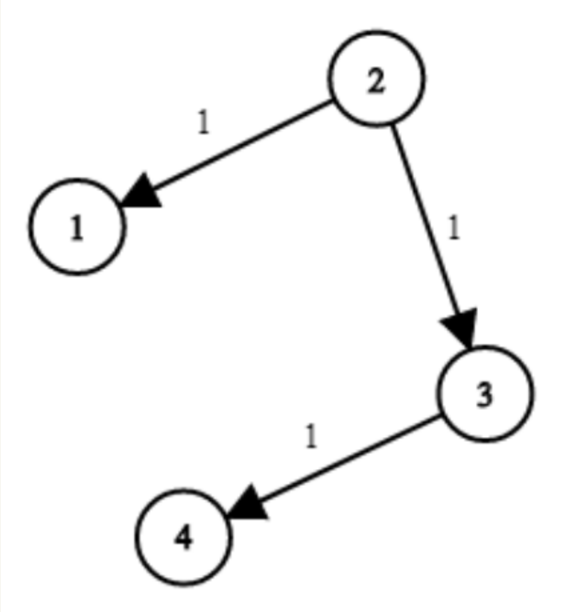
- All Pairs Shortest Path problem


all intermediate vertices{0,1,2.....k-1}
all intermediate vertices{0,1,2.....k-1}
x_1
x_2
i
k
j

Three Layer nested loop: k, i, j

Floyd-Warshall Example

- There are N network nodes, labelled 1 to N. Given times, a list of travel times as **directed** edges times[i] = (u, v, w), where u is the source node, v is the target node, and w is the time it takes for a signal to travel from source to target. Now, we send a signal from a certain node K. How long will it take for all nodes to receive the signal? If it is impossible, return -1.



Input: times = [[2,1,1],[2,3,1],[3,4,1]], N = 4, K = 2
Output: 2