Algorithms

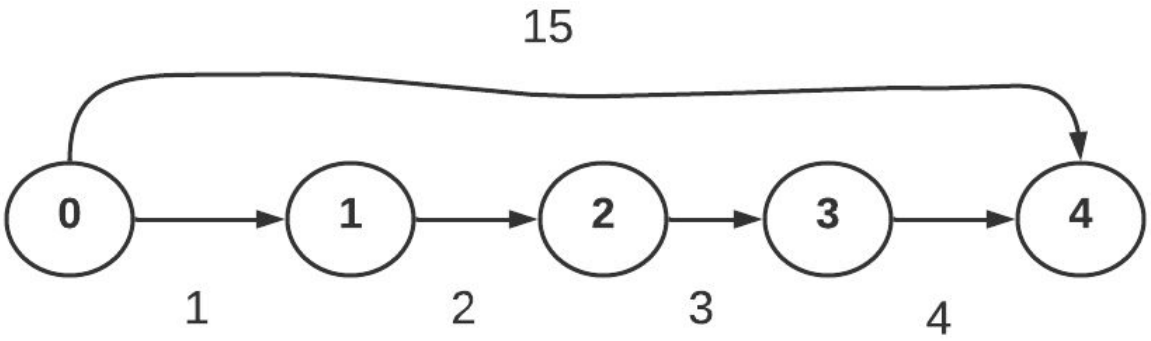
Floyd Homework

Mostafa S. Ibrahim
Teaching, Training and Coaching for more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

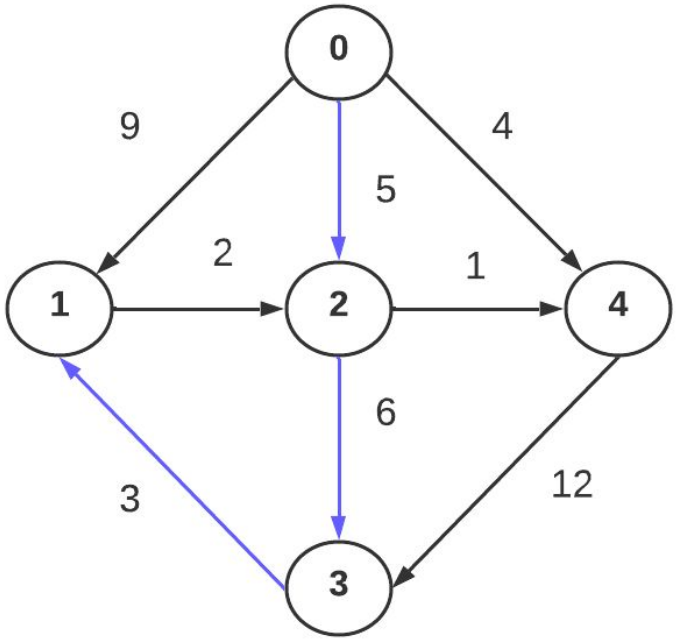
Problem #1: Printing path

- We learned to find the shortest path. Now we would like to print it
- Modify the lecture code and test it to print the path between 2 nodes
- For nodes (0, 4) print:
  - 0 1
  - 1 2
  - 2 3
  - 3 4



Problem #2: Find Minimax Path

- Given a path with weights, a bottleneck(path) = max edge weight
  - If the path has weights [4, 7, 1, 2] then bottleneck = 7
- Assume we have E paths between 2 nodes (i, j)
  - The **minimax** path (i, j) is the path with the **minimum bottleneck** in the E paths
  - That is, the path where the **maximum** edge weight is **minimum**
- Given a directed graph, find the minimax path between every pair of nodes
- Create function compute_min_max(graph)



- There are 4 paths from 0 to 1
  - 0-1: max is 9
  - 0-2-3-1: max is 6
  - 0-2-4-3-1: max is 12
  - 0-4-3-1: max is 12
  - min(9, 6, 12, 12) = 6
- The **minimax** path between (0, 1) has weight 6

Input ⇒ Output

- Read number of nodes N and number of edges M on a line
- Then read M lines, where each has a direct connection between 0-based nodes
    - From To Weight

```
5 8
0 1 9
0 2 5
0 4 4
1 2 2
2 4 1
2 3 6
3 1 3
4 3 12
```

```
0 6 5 6 4
00 0 2 6 2
00 6 0 6 1
00 3 3 0 3
00 12 12 12 0
```

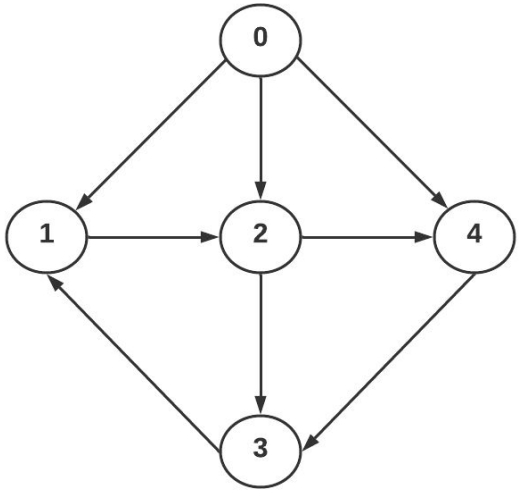
OJ Testing

- Feel free to create test cases, evaluate them using my code, and compare with your own
- If you want to use an OJ, you may use UVA 534 - Frogger
- UVA is an annoying site. Use my provided template file
  - Only implement compute_min_max

Problem #3: Count Paths

- Given a **directed graph**, **counting** how many simple **paths** there are between any 2 nodes.
  - No self-loops
  - M**ultiple edges** between 2 nodes are possible
  - There are maximum of N=30 Nodes
- Print a 2D matrix where O[i][j] = number of paths between i and j
- If there is an infinite number of paths between i and j, prints -1
  - Tip: Due to some **cycle** on some node on the paths

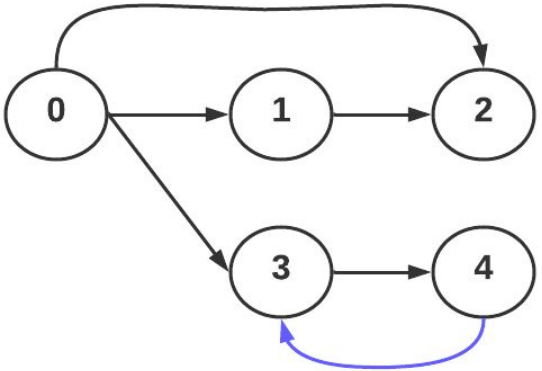
Input ⇒ Output



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

- There are 4 paths from 0 to 1
  - 0-1 / 0-2-3-1 / 0-2-4-3-1 / 0-4-3-1
- There are 3 paths from 0 to 3
  - 0-2-3 / 0-2-4-3 / 0-4-3

Input ⇒ Output




```
0 1 2 -1 -1
0 0 1 0 0
0 0 0 0 0
0 0 0 -1 -1
0 0 0 -1 -1
```

- We have cycle (3-4), but it doesn't affect nodes 1 and 2
- There are 2 paths from 0 to 2
  - 0-1-2 / 0-2
- There are OO paths from 0 to 3
  - 0-3 / 0-3-4-3 / 0-3-4-3-4-3 / …. OO
    - Non-simple paths

Input ⇒ Output

- Read number of nodes N and number of edges M on a line
- Then read M lines, where each has a direct connection between 0-based nodes

```
5 8
0 1 9
0 2 5
0 4 4
1 2 2
2 4 1
2 3 6
3 1 3
4 3 12
```

```
0 6 5 6 4
00 0 2 6 2
00 6 0 6 1
00 3 3 0 3
00 12 12 12 0
```

OJ Testing

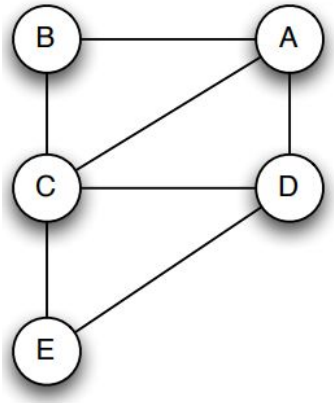
- Feel free to create test cases, evaluate them using my code, and compare with your own
  - See my test cases and their answers in the solution file
- If you want to use an OJ, you may use UVA 125 - Numbering Paths
- UVA is an annoying site. Use my provided **template** file
  - Only implement count_paths(graph)
  - The **graph[i][j]** = number of direct edges between 2 nodes i and j

Hints

- Think how to modify Floyd-Warshall algorithm to find the answer (3 loops)
- Then think about how to identify the cycles
- Then consider how to use the cycles to find the pairs (i,j) with infinite paths

Problem #4: Longest Path

- 1) Consider the longest path problem. For simplicity, use this undirected graph (edge cost = 1). Use ‘Disproof by **counterexample**’ to trivially show that the longest path problem **doesn’t have** the optimal substructure property
  - Find a path. Show that the property is not valid



Problem #4: Longest Path

- 2) What about the longest path in DAG? Does it satisfy the optimal substructure property?
  - What does your answer imply?
- 3) Someone claimed to solve the longest path in a general graph, just multiply all edges by -1, and find the shortest path using Floyd
  - What do you think?

Problem #5: Handling negative cycles

- Someone claimed they have a solution to handle negative cycles
- Find the minimum negative value in the graph (say -3) and add its negative (3) to the whole graph edges.
- Now find the shortest path safely
- Find a counterexample graph to prove this claim is wrong

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."