# LAB 9 : Graph Traversal Simulation & Coding

# [CO3]

## Instructions for students:

- Complete the following methods.
- You may use Java / Python to complete the tasks.
- DO NOT CREATE a separate folder for each task.

## NOTE:

- **YOU CANNOT USE ANY OTHER DATA STRUCTURE OTHER THAN THE ONE MENTIONED IN THE QUESTION.**
- **YOUR CODE SHOULD WORK FOR ANY VALID INPUTS.**

**Python List, Negative indexing and append() is STRICTLY prohibited unless mentioned in the question**

## Lab Tasks: 3 tasks (1~3)

# CODING TASK [Java Template,Python Template]

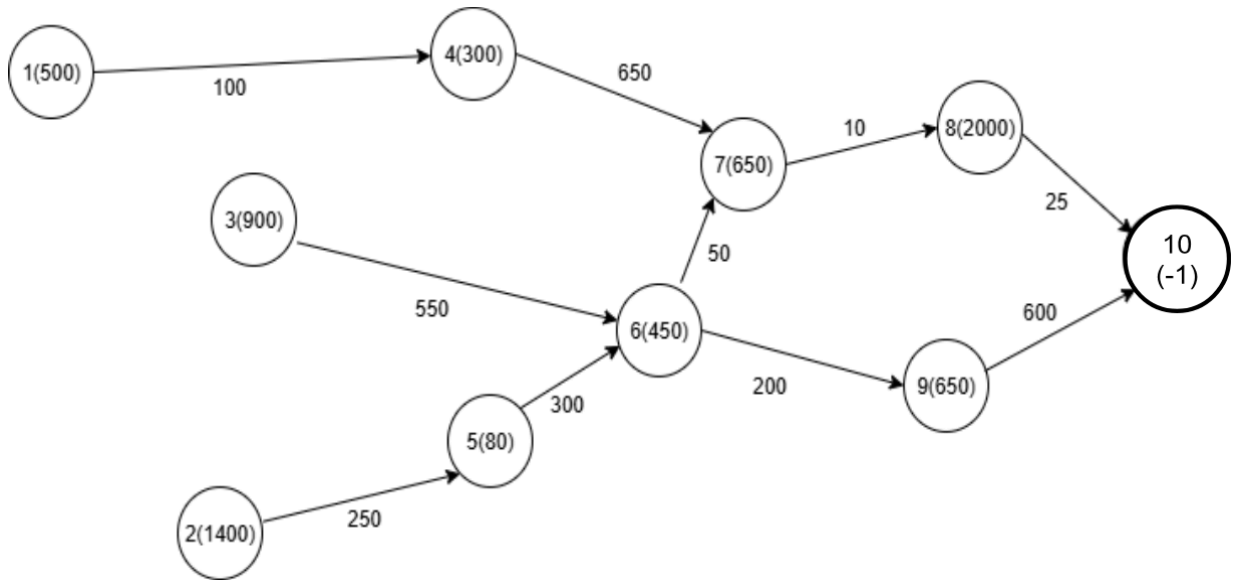
[**Note:** Try solving the coding question using Adjacency Matrix as well]

## Task 1: [Lab Task]

Luffy and Shanks start from given islands with initial power and aim to reach **Laugh Tale (vertex warLord power value is -1)**. Between islands, they fight a **Marine Officer (power given as edge weight)**. On each island, they fight a **Warlord (power given as vertex value)**. They always choose the next island with the **minimum (Marine Officer + Warlord) power**. Furthermore, stamina decreases after every single fight. If a pirate's stamina becomes less than the Warlord's power at any island, they cannot continue and fail.

**Winner rules:**

- If only one of them reaches Laugh Tale and the other can't → the one who reaches wins.
- If both reach Laugh Tale → the one with higher remaining stamina wins.
- If none reach Laugh Tale → Winner is no one.




**[You can assume there's won't be any loops from the starting point to Laugh Tale]**

*[You can take helper functions if needed]*

| Python Method Signature |
|---|
| `def laughTale(wL, adjL, sSt, sP, lSt, lP):`<br>`  #TODO` |
| **Java Method Signature** |
| `public static void laughTale(Integer[] warLords, Edge[] adjL, int sSt, int sP, int lSt, int lP){`<br>`  //TODO`<br>`}` |

**Warlord Powers, Sample Graph & Edge Class**

**warLords powers :** [null, 500, 1400, 900, 300, 80, 450, 650, 2000, 650, -1]

**Given Adjacency List :**

0 : null
1 : <4,100> -> null
2 : <5,250> -> null
3 : <6,550> -> null
4 : <7,650> -> null
5 : <6,300> -> null
6 : <7,50> -> <9,200> -> null
7 : <8,10> -> null
8 : <10,25> -> null
9 : <10,600> -> null
10 : null

**Given Java Edge Class**

```
class Edge{
       int toV, weight;
       Edge next;
       public Edge(int t, int w){
              this.toV = t;
              this.weight = w;
       }
}
```

**Given Python Edge Class**

```
class Edge:
    def __init__(self, toV, weight):
        self.toV = toV
        self.weight = weight
        self.next = None
```

| Sample Starting Vertex and Graph | Output |
|---|---|
| **sSt = 2, sP = 7000, lSt = 1, lP = 3000** | Winner is Shanks |

**Explanation**

**sSt:** Shank's start from Island: 2
**sP:** Shanks's power: 7000
**lSt:** Luffy starts from Island: 1
**lP:** Luffy's power: 3000

**Luffy's path:** vertex 1 -> vertex 4 -> vertex 7 -> vertex 8
**Luffy's power level:** 3000-500-100-300-650-650-10=790 *[no more power left to reach vertex 10]*
**Shanks's path:** vertex 2 -> vertex 5 -> vertex 6 -> vertex 7 -> vertex 8 -> vertex 10
**Shanks's power level:** 7000-1400-250-80-300-450-50-650-10-2000-25= 1785.
So, Luffy cannot reach Laugh Tale and the **Winner is Shanks.**

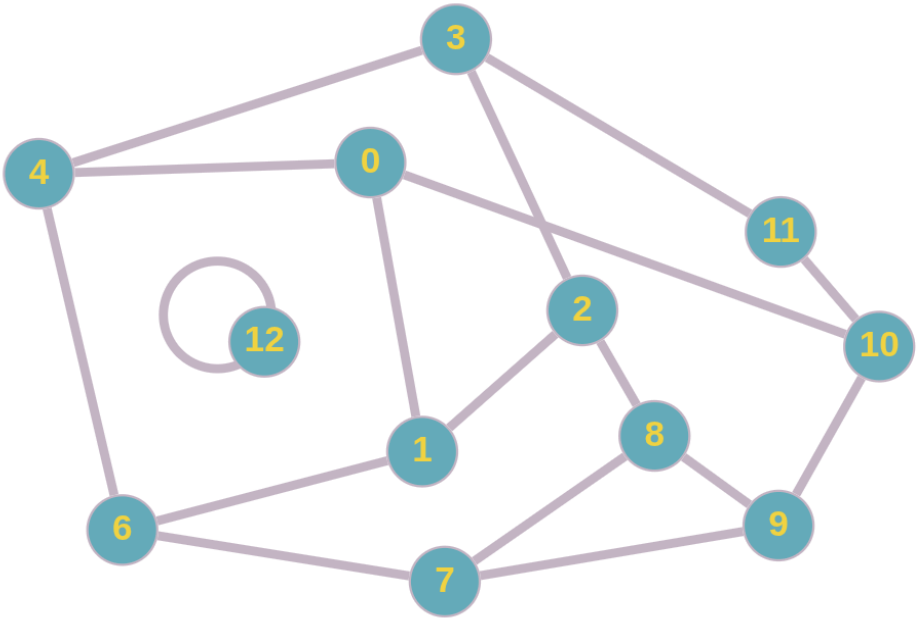
| Sample Starting Vertex and Graph | Output |
|---|---|
| **sSt = 2, sP = 7000, lSt = 3, lP = 6500** | Winner is Luffy |

**sSt:** Shank's start from Island: 2
**sP:** Shanks's power: 7000
**lSt:** Luffy starts from Island: 3
**lP:** Luffy's power: 6500

**Luffy's path:** vertex 3 -> vertex 6 -> vertex 7 -> vertex 8 -> vertex 10
**Luffy's power level:** 6500-900-550-450-50-650-10-2000-25=1865
**Shanks's path:** vertex 2 -> vertex 5 -> vertex 6 -> vertex 7 -> vertex 8 -> vertex 10
**Shanks's power level:** 7000-1400-250-80-300-450-50-650-10-2000-25= 1785
Since 1865>1785. So, the **winner is Luffy.**

# SIMULATION TASKS:

## Task 2: [Lab Task]




**Simulate the BFS algorithm using a Queue** choosing any specific vertex as a starting point.

## Task 3: [Lab Task]

**Simulate the DFS algorithm (including discovery time & finishing time)** choosing any specific vertex as a starting point.