
# ESLogistic - Logistics Plan Application

## Group Project - ICE3402P Data Structure

HAN Jiaming (韩佳明), SUN Shi (孙石), XU Wenxin (徐文欣), ZHU Hongao (朱虹翱).

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY


# ESLogistic = Application + Algorithm

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

# Agenda

## ESLogistic - Logistics Plan Application

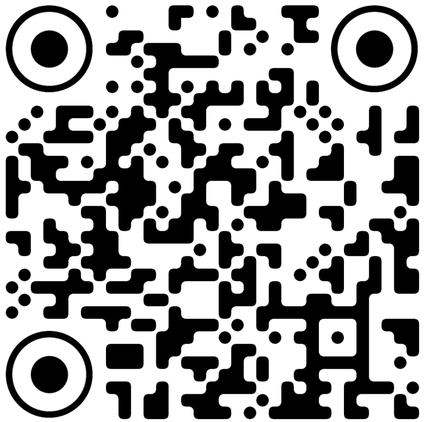
- Preview and Demo
- Application Architecture
- Algorithm Details
  1. Improved Bellman-Ford
  2. Reinforcement Learning
  3. Dijkstra Algorithm





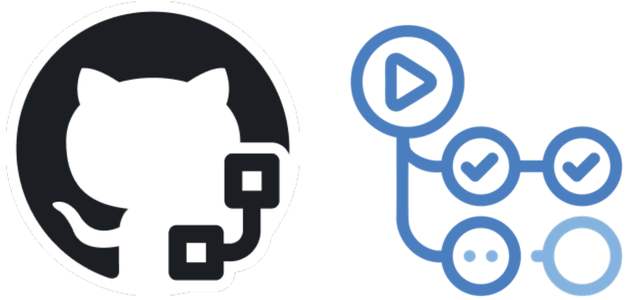




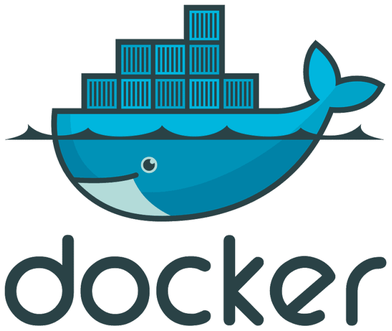


# Preview & Demo

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY



github.com/guomaimang/ESlogistic

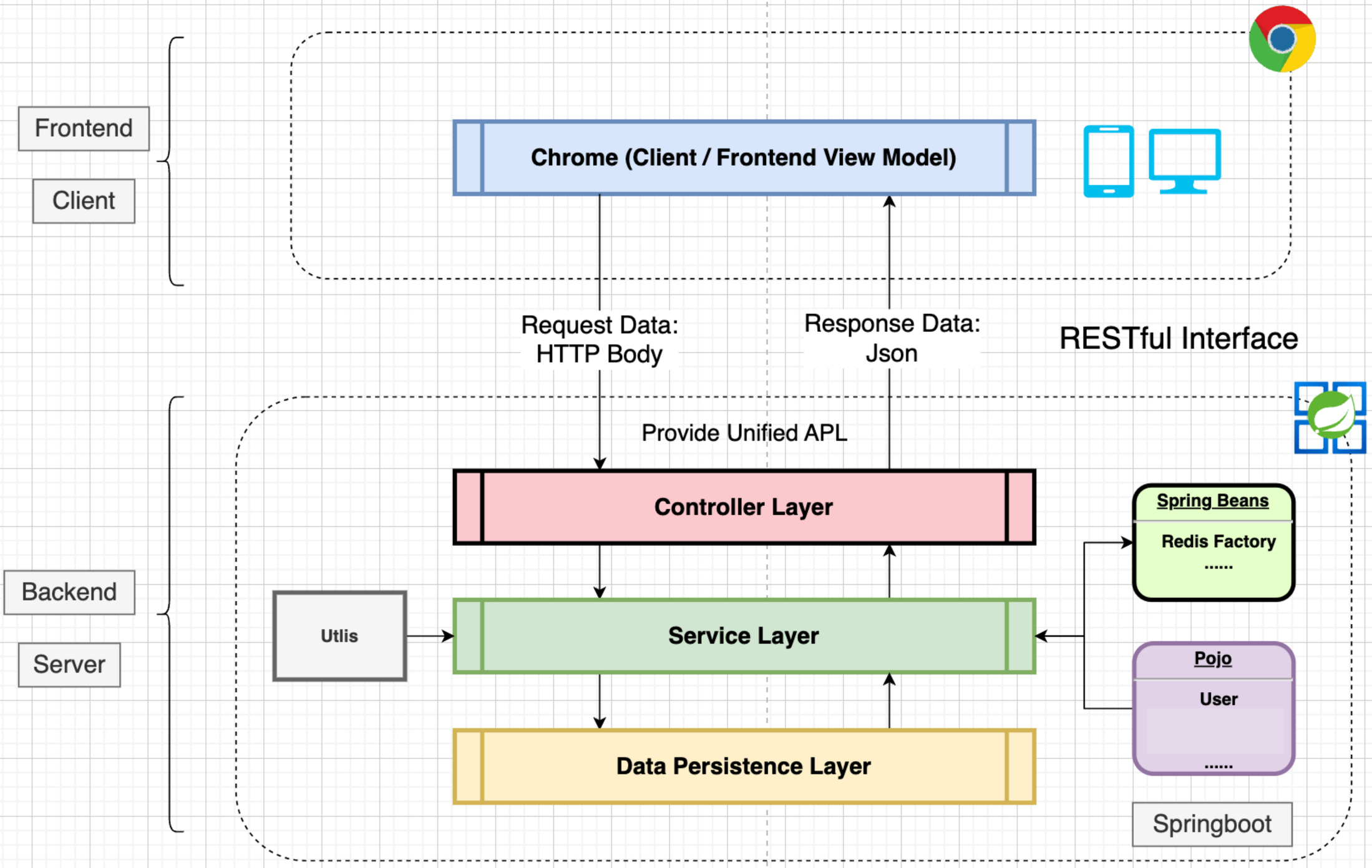


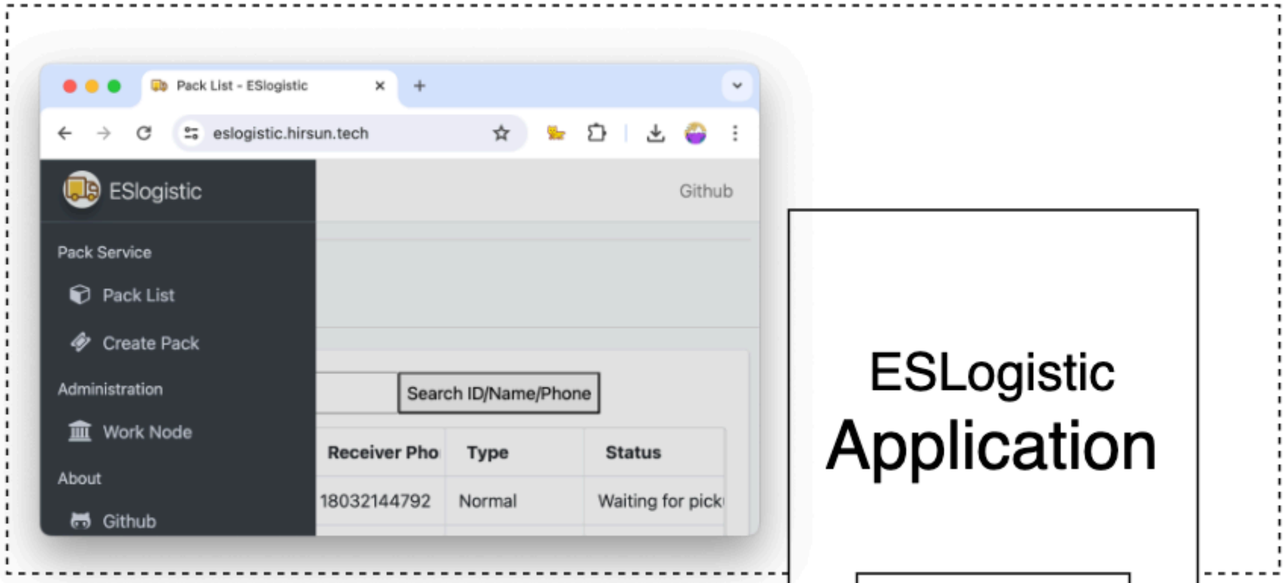

https://eslogistic.hirsun.tech/

**Publicly Available, Not Limited to Theory!**

# Application Architecture

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY


Frontend
Client
Chrome (Client / Frontend View Model)
Request Data:
HTTP Body
Response Data:
Json
RESTful Interface
Provide Unified APL
Backend
Server
Controller Layer
Utlis
Service Layer
Data Persistence Layer
Spring Beans
Redis Factory
.....
Pojo
User
.....
Springboot







# ESLogistic = Application + Algorithm S

```java
public interface RouterService {

    public List<String> route (Map stationsBase, Map StationsPackAmount,
                               Map RoutesPackAmount, String packFrom, String packTo, int packPriority);

    public WorkNode nextHop(Pack pack);

    public void initService();
}
```

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

上海交通大學

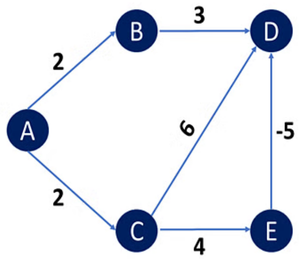
SHANGHAI JIAO TONG UNIVERSITY

PART TWO

# Improved Bellman-Ford

# Improved Bellman-Ford

- Abstract modeling
  - Node: The station or the center
  - Weighted Edge: The cost in the route or the station or the center
  - The shortest path: Use of Bellman-Ford
- Pros and cons
  - Stable
  - Lack of predictability


| | B | C | D | E |
|---|---|---|---|---|
| 0 | ∞ | ∞ | ∞ | ∞ |
| 0 | 2 | 2 | ∞ | ∞ |
| 0 | 2 | 2 | 2 | 6 |
| 0 | 2 | 2 | 3 | 6 |
| 0 | 2 | 2 | 3 | 6 |

上海交通大學

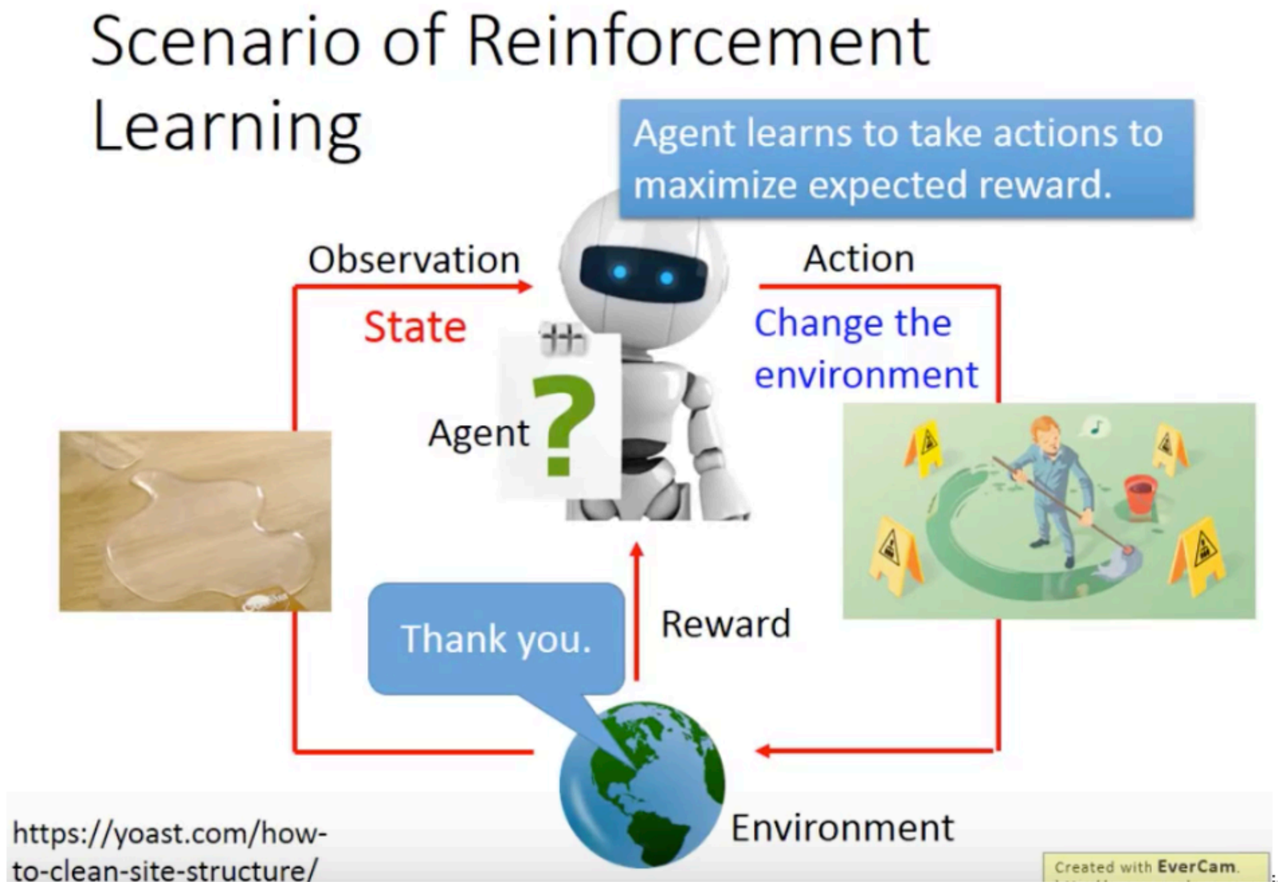
SHANGHAI JIAO TONG UNIVERSITY

# PART THREE

# Reinforcement Learning

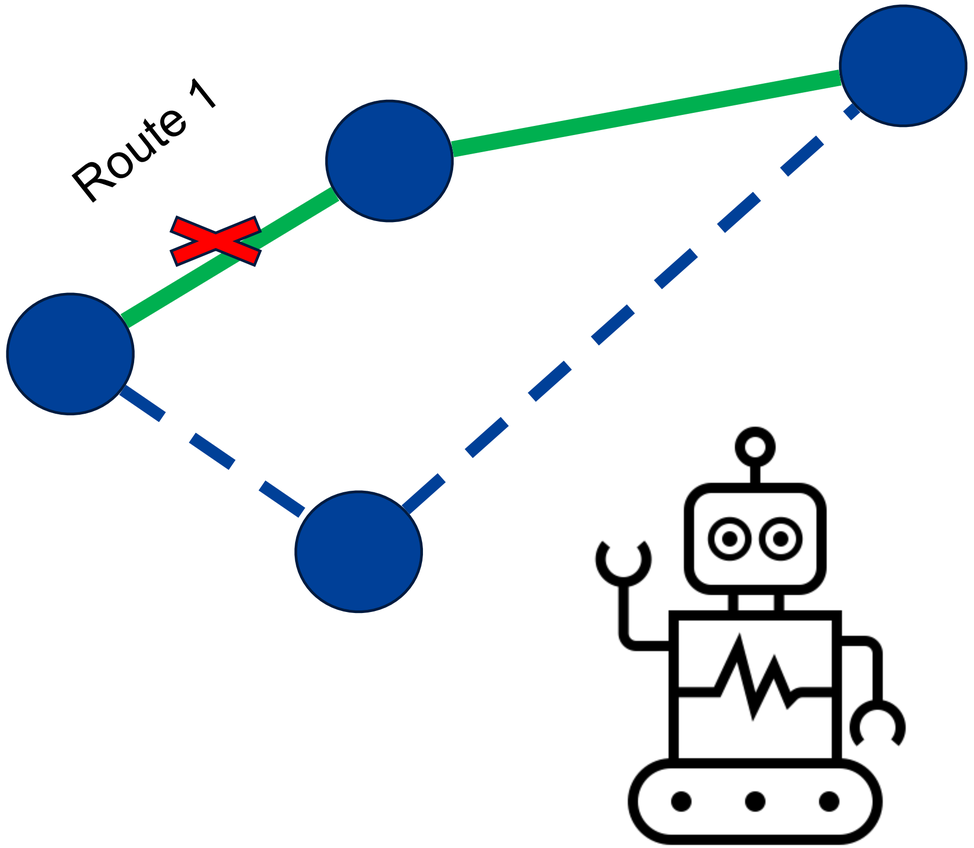
# Reinforcement Learning

- Environment (Logistics Simulator):
  - Step: update TimeTick
  - State: num_pack at each route/node
  - Reward: undone packs
  - End state: all packs are delivered
- Agent (Route switches)
  - Choose action: which routes to temper
  - Train: using DQN
  - Default policy: Improved Bellman Ford



# Reinforcement Learning

- Environment (Logistics Simulator):
  - Step: update TimeTick
  - State: num_pack at each route/node
  - Reward: undone packs
  - End state: all packs are delivered
- Agent (Route switches)
  - Choose action: which routes to temper
  - Train: using DQN
  - Default policy: Improved Bellman Ford




**“Close Route 1!”**

上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

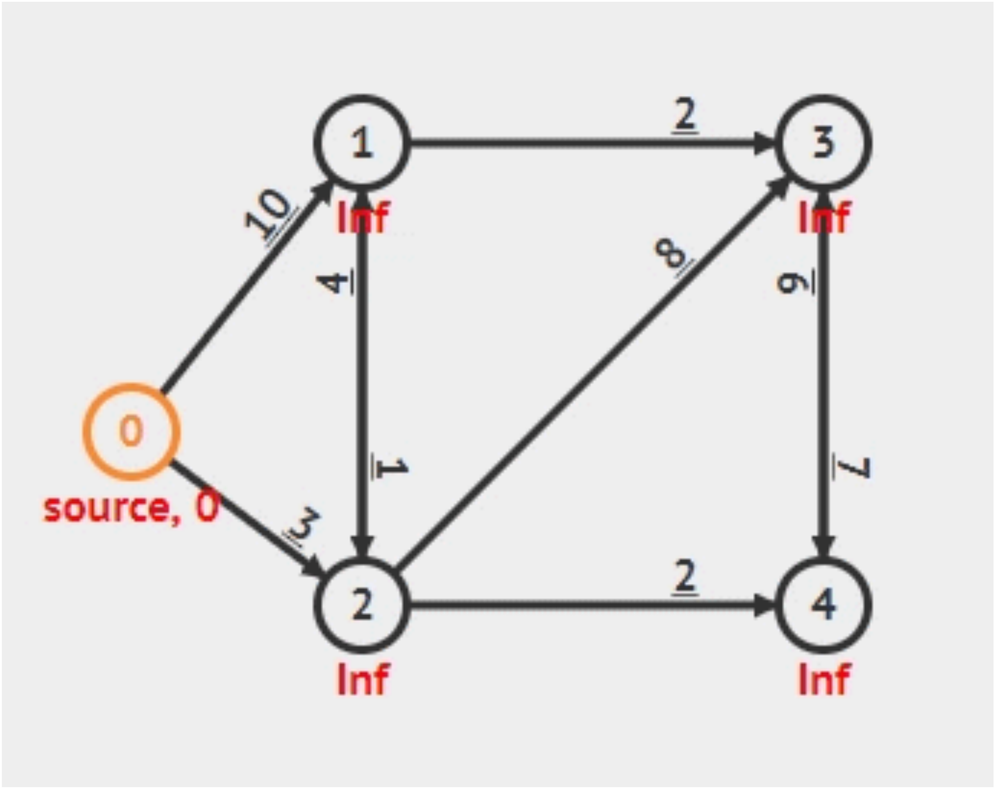
# PART FOUR

# Dijkstra Algorithm


# Dijkstra Algorithm

## Basic Theorem:

- Graph: two dimensional array adjacency matrix and initialized.
- Choose a vertex: starting point.
- Construct a d one-dimensional array: **dis[n]** *(**n**: the number of vertices)( **dis**: record the shortest path distance)*
- Find the minimum value from the dis array each time. Add new points, update dis array *(distance: short(**yes**) or long(**no**))*
- Repeat update: all points find the shortest path *(**end**)*



# *Thank You*

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學

Faculty of Science and Technology
University of Macau