

# 全點對最短路

# 全點對最短路

- 所有點對的最短路徑
- $V$  次單源最短路
- Floyd Warshall

# Floyd-Warshall

## 理念

- DP 它！
- 狀態： $dp[k][i][j]$ ：只能使用  $1 \dots k$  作為中繼點， $i$  到  $j$  的最短路徑
- 轉移： $dp[k][i][j] = \min(dp[k-1][i][j], dp[k-1][i][k] + dp[k-1][k][j])$
- 基底：
  - $dp[0][i][i] = 0$
  - $dp[0][i][j] = w[i][j]$
- 若  $i, j$  沒有邊， $w[i][j] = INF$

# Floyd-Warshall (cont.)

## 程式碼

```
int w[N][N], dis[N][N];

void floyd_warshall()
{
    for (int i=1; i<=n; i++)
        for (int j=1; j<=n; j++)
            dis[i][j] = (i==j ? 0 : w[i][j]);

    for (int k=1; k<=n; k++)
        for (int i=1; i<=n; i++)
            for (int j=1; j<=n; j++)
                dis[i][j] = min(dis[i][j], dis[i][k] + dis[k][j]);
}
```

# Floyd-Warshall (cont.)

複雜度

- $O(V^3)$

# Lab16: Shortest Routes

- Given  $n$  vertices and  $m$  undirect edges.
- $q$  queries of shortest path between  $a$  and  $b$
- $n \leq 500$
- $m \leq n^2$
- 直接使用 Floyd Warshal 預先找好最短路

# Lab16: Shortest Routes

## AC Code (URL)

```
const int N = 505;

int n, m, q;
long long dis[N][N];

int main() {
    ios::sync_with_stdio(0), cin.tie(0);

    cin >> n >> m >> q;

    for (int i = 1; i <= n; i++)
        for (int j = 1; j <= n; j++)
            dis[i][j] = (i==j? 0 : 1e18);

    while (m--) {
        long long u, v, w;
        cin >> u >> v >> w;
        dis[u][v] = min(dis[u][v], w);
        dis[v][u] = min(dis[v][u], w);
    }
```

```
    for (int k=1; k<=n; k++) {
        for (int i = 1; i<=n; i++){
            for (int j = 1; j<=n; j++) {
                dis[i][j] = min(dis[i][j], dis[i][k] + dis[k][j]);
            }
        }
    }

    while (q--) {
        int u, v;
        cin >> u >> v;
        cout << (dis[u][v] >= 1e18? -1 : dis[u][v]) << '\n';
    }
}
```