

# VE477 Lab5

---

于欣淼 YU Xinmiao 518021910792

## Q1

The implementation of sparse graph and dense graph is in `code/graph.py` as `class SGraph` and `class DGraph` respectively. The main difference is that store the information of edge or vertex in the class.

## Q2

Implement in `code/Dijkstra.py` with Fibonacci Heap in `code/FibonacciHeap.py`.

## Q3

Implement in `code/BellmanFord.py`.

## Q4

- Complexity

For *BELLMAN-FORD*, initialize the graph takes  $\Theta(|V|)$ , as each vertex will take  $\Theta(1)$ . Then pass over all the edges takes  $\Theta(|VE|)$ . Then check if a negative cycle exists takes  $\Theta(E)$ . Totally  $\mathcal{O}(VE)$ .

For *Dijkstra*, as the min-priority queue is implemented through *Fibonacci Heap*, the amortized cost of each of the  $|V|$  *EXTRACT-MIN* operation is  $\mathcal{O}(\log V)$ , for each of the *DECREASE-KEY*, happens at most  $E$ , is  $\mathcal{O}(1)$ . So totally  $\mathcal{O}(V \log V + E)$ .

- Running time

As I fail some JOJ cases of *Dijkstra*, the time is recorded only when find the output of these two algorithm is same

