

数据结构与算法作业 3

2022 年 6 月 16 日

一、图 $G=(V, E)$ ，用 Johnson 算法，计算所有结点对之间的最短路径。Johnson 算法伪代码如下：

```
JOHNSON( $G, w$ )
1  compute  $G'$ , where  $G'.V = G.V \cup \{s\}$ ,
    $G'.E = G.E \cup \{(s, v) : v \in G.V\}$ , and
    $w(s, v) = 0$  for all  $v \in G.V$ 
2  if BELLMAN-FORD( $G', w, s$ ) == FALSE
3      print "the input graph contains a negative-weight cycle"
4  else for each vertex  $v \in G'.V$ 
5      set  $h(v)$  to the value of  $\delta(s, v)$ 
       computed by the Bellman-Ford algorithm
6  for each edge  $(u, v) \in G'.E$ 
7       $\hat{w}(u, v) = w(u, v) + h(u) - h(v)$ 
8  let  $D = (d_{uv})$  be a new  $n \times n$  matrix
9  for each vertex  $u \in G.V$ 
10     run DIJKSTRA( $G, \hat{w}, u$ ) to compute  $\hat{\delta}(u, v)$  for all  $v \in G.V$ 
11     for each vertex  $v \in G.V$ 
12          $d_{uv} = \hat{\delta}(u, v) + h(v) - h(u)$ 
13     return  $D$ 
```

1. 使用斐波那契堆来实现 Dijkstra 算法里面的最小优先队列；
2. 使用最小二叉堆来实现 Dijkstra 算法里面的最小优先队列；
3. 分别给出稀疏图、稠密图作为输入进行测试，分析两种实现方法的时间复杂度和空间复杂度。
4. （附加题）设计时间复杂度比 Johnson 算法更小的算法，并给出源代码。（选做）

要求：

1. 所有代码使用 C++ 编写，并给出关键代码的注释。
2. 测试案例的规模大于（邻接矩阵） 100×100 。