

# 作业八：Bellman-Ford 算法的实现

李沁霞

3210300363 统计学

2022 年 12 月 13 日

## 1 简介

Bellman-Ford 算法的实现与其他动态规划问题一样，该算法以自下而上的方式计算最短路径。它首先计算路径中最多有一条边的最短距离。然后，他计算多有 2 条边的最短路径，以此类推。在外循环的第  $i$  次迭代之后，计算最多  $i$  条边的最短路径。可以有最大值  $|V|$  任何简单路径中的 1 个边，这就是外循环运行  $|v|$  的原因-1 次。

下面是算法的步骤：

1. 让给定的源顶点为 0。除了源点距离，将所有距离初始化为无限。
2. 让所有边按照顺序处理。当第一次处理所有边时，我们得到初始距离。再次处理所有边，当不能改变所有边的距离才完成迭代。

## 2 测试结果

V	500	1000	2000	4000
$E = 2* V $	0.014	0.05	0.268	1.049
$E = 4* V $	0.019	0.101	0.386	1.871
$E = 8* V $	0.024	0.15	0.636	4.646
$E = 16* V $	0.07	0.341	2.039	21.845

```
==3264== Memcheck, a memory error detector
==3264== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==3264== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==3264== Command: test
==3264==
==3264==
==3264== HEAP SUMMARY:
==3264==   in use at exit: 0 bytes in 0 blocks
==3264==   total heap usage: 30 allocs, 30 frees, 4,065 bytes allocated
==3264==
==3264== All heap blocks were freed -- no leaks are possible
==3264==
==3264== For lists of detected and suppressed errors, rerun with: -s
==3264== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```