

dijkstra 算法说明文档。

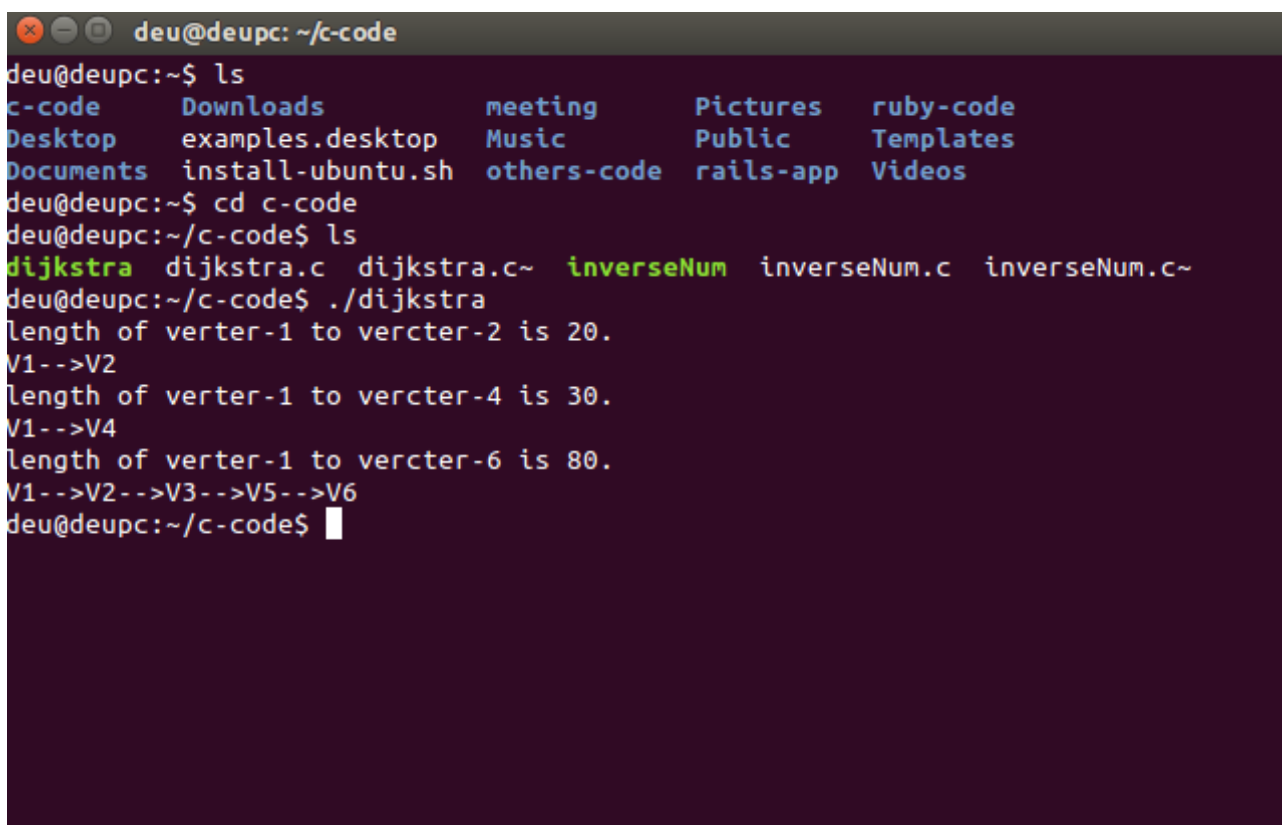
1. 原算法求出了单点源到其他各点的最短路径长度，算法时间复杂度是 n 的平方。本程序用 C 语言来描述这个算法，各功能块通过函数定义和调用实现，代码可重用性比较高。特别指出，为了免去函数调用时候参数传递的麻烦，把耗费邻接矩阵二位数组定义成了全局变量，尽管这样导致安全性降低。算法注释丰富，可读性很好。

2. 对原算法作两点修改来回溯路径

2.1 定义数组 parent 来存储每个节点的最短路径先驱节点。

2.2 利用先驱数组 parent 来求取最短路径，这部分参考了别人博客上的代码。这部分代码时间复杂度是 n ，没有增加原来的时间复杂度，算法总的时间复杂度是 n 的平方，算法可行。

附图：测试结果



```
deu@deupc: ~/c-code
deu@deupc:~$ ls
c-code      Downloads  meeting    Pictures    ruby-code
Desktop     examples.desktop  Music      Public      Templates
Documents   install-ubuntu.sh others-code rails-app   Videos
deu@deupc:~$ cd c-code
deu@deupc:~/c-code$ ls
dijkstra  dijkstra.c  dijkstra.c~  inverseNum  inverseNum.c  inverseNum.c~
deu@deupc:~/c-code$ ./dijkstra
length of verter-1 to vercter-2 is 20.
V1-->V2
length of verter-1 to vercter-4 is 30.
V1-->V4
length of verter-1 to vercter-6 is 80.
V1-->V2-->V3-->V5-->V6
deu@deupc:~/c-code$
```

互动交流

1. word或pdf文件，包含对新算法的描述以及试验结果的截图
2. 源文件