



算法设计与分析

实验报告（四）实现SPA算法

姓 名	熊恪峥
学 号	22920202204622
日 期	2022年6月1日
学 院	信息学院
课程名称	算法设计与分析

实验报告（四）实现SPA算法

目录

1	问题描述	3
2	实现思路	3
3	算法分析	3
4	实现代码	3

1 问题描述

实现最短路增广算法解决最大流问题。

2 实现思路

Dinic算法是一种强多项式时间复杂度的算法，用来计算最大流。它的时间复杂度是 $\mathcal{O}(VE^2)$ 。这种算法在1970年由以色列计算机科学家Yefim A. Dinitzf提出，它优化了时间复杂度为 $\mathcal{O}(V^2E)$ 的Edmond-Karp算法。它的关键在于根据层级图进行增广路的寻找，如算法1。

算法 1 Dinic算法

```
1: procedure DINIC( $G$ )
2:   for each  $e \in E$  do
3:      $f(e) = 0$ 
4:   Compute  $G_L$  from  $G_f$ 
5:   while there is a path to  $t$  in  $G_L$  do
6:     find a blocking flow  $f' \in G_L$ 
7:     augmenting flow  $f$  by  $f'$ 
8:   Compute  $G_L$  from  $G_f$ 
```

3 算法分析

计算 G_L 需要进行广度优先搜索，它的时间复杂度是 $\mathcal{O}(E)$ ，在 G_L 中寻找阻塞流需要 $\mathcal{O}(VE)$ 因此Dinic算法的时间复杂度是 $\mathcal{O}(VE^2)$ 。

使用动态树的高级数据结构可以进一步将寻找阻塞流的时间复杂度缩减到 $\mathcal{O}(\log VE)$ ，时间复杂度将会降低到 $\mathcal{O}(VE \log V)$ 。

4 实现代码

代码 1: 红包发放

```
1
2 import random
3 from typing import Final, List
4
5 scheme_diff: List[int] = list({1.65, 1.67, 16.79, 1.77, 17.79, 1.87, 18.79, 1.98,
6     5.19, 0.65, 6.59, 6.65, 0.07, 0.87, 8.79, 8.87, 0.98, 9.89, 9.98})
7
8 m, n = input("m and n:").split()
9
10 n = int(n)
11 m = float(m) - n * 0.01
12
13 result = list([0.01 for i in range(0, n)])
14
15 for i in range(0, n):
16     amount = scheme_diff[random.randrange(0, len(scheme_diff))]
17     if m - amount == 0:
18         result[i] += amount
```

```
19         break
20     elif m - amount < 0:
21         result[i] += m
22         break
23     else:
24         m -= amount
25         result[i] += amount
26
27 result = list([ round(r, 2) for r in result])
28
29 print(result)
```