# 算法设计 实验三

## 1.实验内容：实现preflow-push算法求解有向图的最大流。

## 2.实验目的：理解流的概念和最大流的概念；理解preflow-push的基本思想；理解preflow-push和ford-fulkerson两个算法的相同之处与区别。

## 3.实验要求：设计并实现算法代码；完成并提交实验报告。

## 问题求解过程：

1. 理解算法思想：

该算法最本质的想法就是想尽可能多地往outgoing的边上输出flow（push），然而每一个中间节点最后可以流出的flow是有限的，就会使得一些节点有一些flow流不出去，所以我们就要想办法让这些节点里面的flow流回去，这个时候就需要引入高度（H）的概念，flow只可以从高处流向低处，通过高度的调整（relabel）使得含有多余流量的节点可以把flow往回流。

1. 设计算法步骤：

首先从起点开始做头节点的push操作，输出头节点的outgoing的边的权值的流量，push\_start();

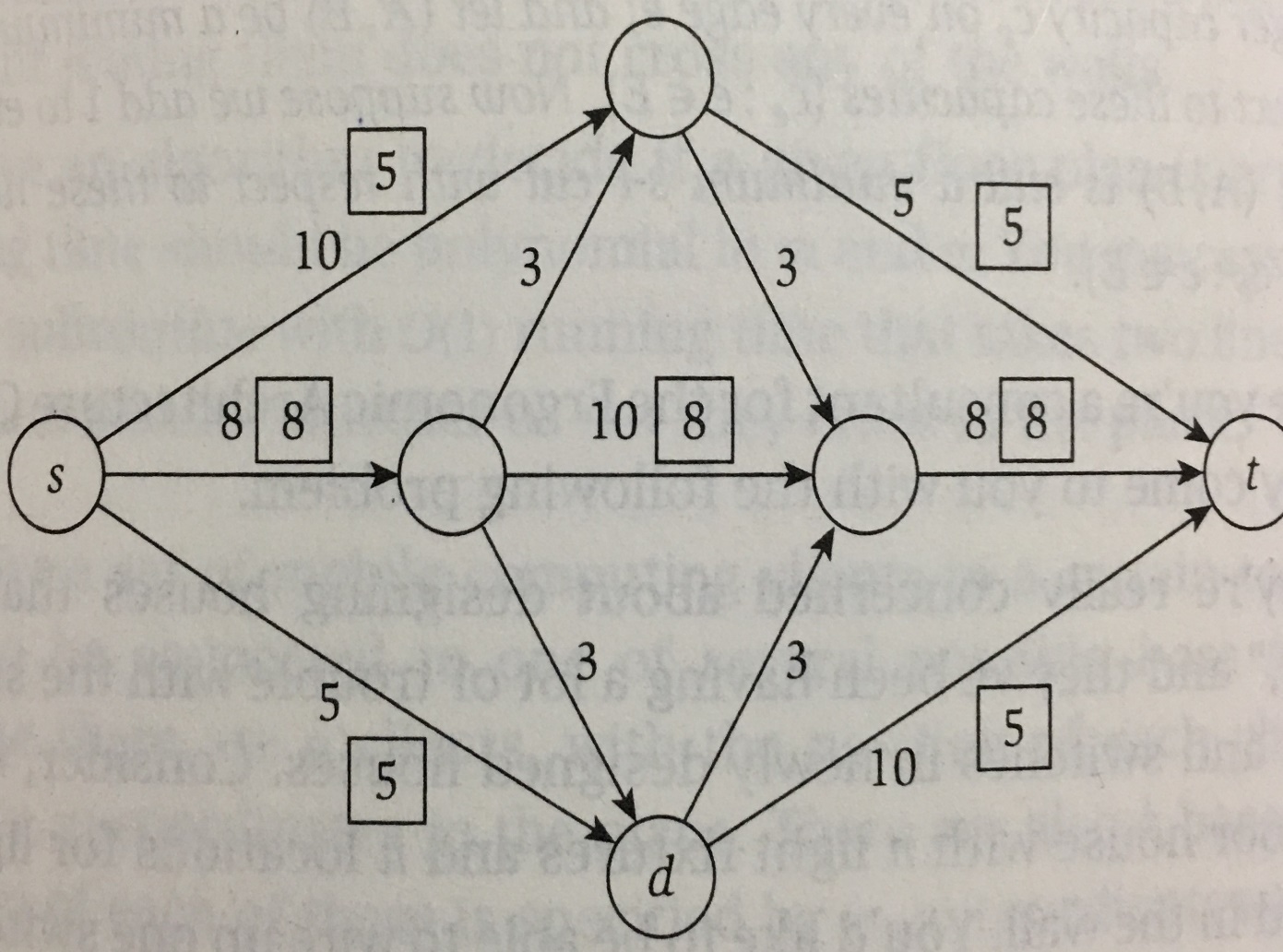
然后开始整个push操作和relable操作的循环模式，直到图中不存在含有多余流量的点为止。preflow\_push()；push(i); relabel(i);

## 存储信息的数据结构：

1. 存储有向图(Directed Graph)的边长的数据结构：int G[MAX][MAX]
2. 存储剩余流（residual graph）的图的数据结构：int R[MAX][MAX]
3. 存储活性点（excess point）的数据结构：int E[MAX]
4. 存储点的高度 (height) 的数据结构 : int H[MAX]
5. 存储图中点的个数（n）：int n

## 程序代码以及运行结果：

1. 源代码：见附属文件。
2. 测试：



以该图为样例，输入：

6

0 1 10

0 2 8

0 3 5

1 4 3

1 5 5

2 1 3

2 4 10

2 3 3

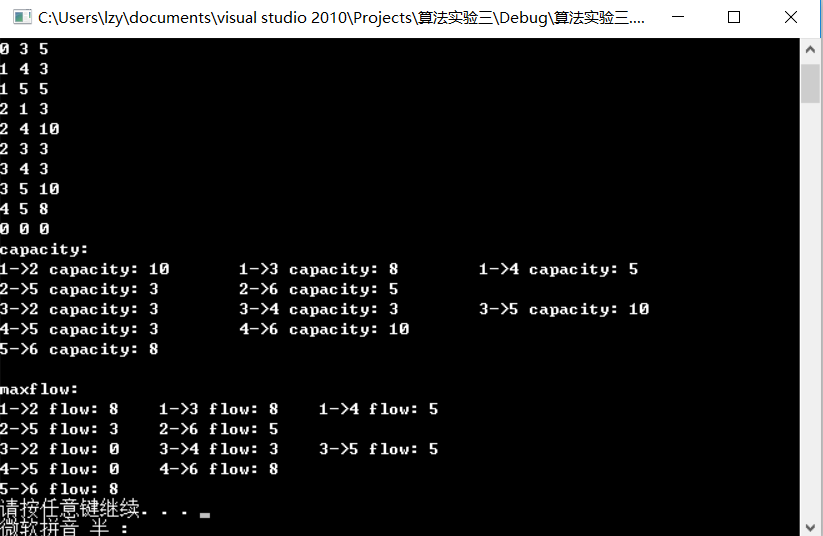
3 4 3

3 5 10

4 5 8

0 0 0

程序运行结果：



所以得到最大流的结果。