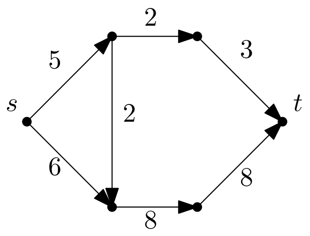
算法设计与分析 LECTURE 12

第十三周作业-solution

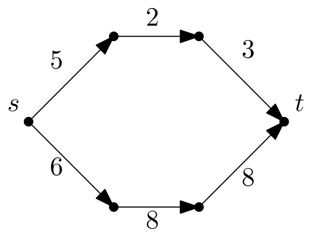
LECTURER: 杨启哲 LAST MODIFIED: 2023 年 12 月 26 日

1. (教材习题 15.12) 用例子说明, 在剩余图的层次图中的一个阻塞流并不一定是剩余图中的阻塞琉。

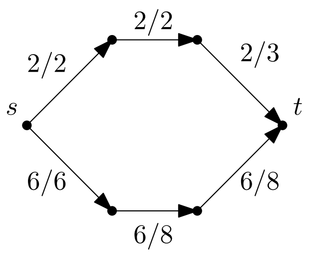
证明. 我们考虑如下的例子:



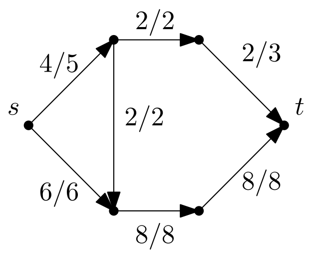
G_1



G_2



G_3



G_4

上图 G_1 是一个流网络，边上的值代表了相应的容量限制。特别的我们令流量 f 为 0，从而此时的剩余图就是 G_1 本身。

G_2 是其相应的层次图。不难发现，G_3 给出了其上的阻塞流，但很显然该阻塞流不是剩余图 G_1 上的阻塞流，其阻塞流如 G_4 所示。主要原因在于，层次图中只反映了当前剩余图中最短路径的增广路径信息。 □

2. (教材习题 15.44) 给定有向图 $G = (V, E)$ 和其中两个特殊的顶点 s, t。试设计一个有效算法，找出 s 到 t 的边不相交的路径的最大条数。

证明. 这个问题可以简单的转换成最大流问题。构造如下的一个流网络 $G = (V, E, c)$，不需要对点和边做任何改动，只需要对每条边的容量限制设为 1。则我们有如下事实：

- s 到 t 的边不相交的路径的最大条数等于 G 中的最大流的值。

只要注意到，一条 s 到 t 的路径是并且只可能是一个容量为 1 的流，而两条不相交的路径的流可以叠加即可获得上述结论。

从而我们在流网络上运用最大流算法求解最大流即可。 □

3. 给定无向图 $G=(V,E)$，称其能被 2 着色，如果存在一个染色方案将所有顶点都染成黑白两色，使得如果两个点相邻，则这两个点的颜色不同。试设计一个有效算法，判断给定的无向图能否被 2 着色。

证明. 这一问题实际上就是判断一个无向图是否是二分图的问题。因此与第 6 次作业的第 2 题解法相同：

二分图的充要条件时没有奇圈，因此我们只需要判断图中是否存在奇圈即可，这可以通过在 DFS 的过程中对点交替染色来实现，如果碰到一个已经染色过的顶点，那么如果颜色相同则说明存在奇圈，返回 $false$，否则继续 DFS。算法流程如下：

判断二分图

输入: 图 G
输出: G 是否是二分图

```
color[v] = 1
DFS(G, v)
return true
function DFS(G, v)
    visited[v] ← true
    for u ∈ G.adj[v] do
        if not visited[u] then
            color[u] ← 1 − color[v]
            DFS(G, u)
        else if color[u] == color[v] then
            return false
        end if
    end for
end function
```

算法所需时间即为 DFS 的搜索时间，即 $O(|V|+|E|)$。 □