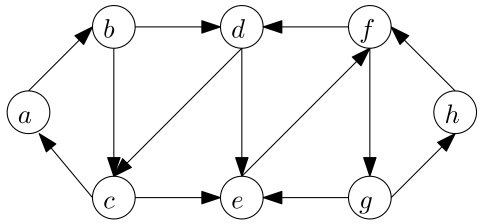
算法设计与分析 LECTURE 6

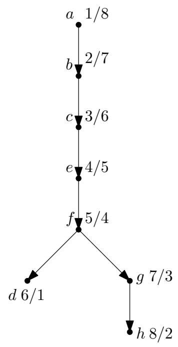
第七周作业-solution

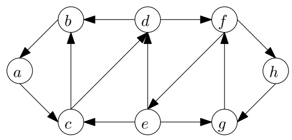
LECTURER: 杨启哲 LAST MODIFIED: 2023 年 12 月 24 日

1. (教材习题 8.16) 请在下图所示的有向图上应用强连通分支的算法。



解答.





- 首先从 a 出发可以得到对应的一颗 DFS 树如上左图所示，其 post 序按照从小到大的顺序为 a, b, c, e, f, g, h, d。
- 根据这个顺序在原图的逆图 (上右图) 中进行 DFS，注意到在这个例子中从 a 出发可以访问到所有的点，因此原图整个图是一个强连通分支。

□

2. (教材习题 8.31) 二分图是指，图的顶点集合可以被分为两个互不相交的子集，使得每条边的两个顶点分别属于这两个子集。请给出一个有效的算法，判断一个图是否是二分图。

解答. 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 $O(|V|+|E|)$。 □

3. (教材习题 7.18) 请指出，当图中存在负边的时候，算法 $Dijkstra$ 的正确性证明哪里存在问题导致失效。

解答. 在 $Dijkstra$ 的正确性证明中的归纳部分的如下不等式中：

$$\omega(1,w) \leq \omega(1,y)$$

我们利用了到达 y 的最短路径的长度一定比到达在这条路径中的一个点 w 的长度长这一性质,但很遗憾当有负权重的时候这一性质不再成立，从而这个正确性证明对于负权重是不成立的。 □

4. (DAG 中的路径) 请给出一个算法计算给定一个有向无环图中所有的路径数量，并分析你的算法。

解答. 基本思路是先将 DAG 输出其拓扑序列，假设其拓扑序为 $v_1,\ldots,v_k$，定义 $P(i,j)$ 为 v_i 到 v_j 的路径数量，则我们有：

$$P(i,j) = \begin{cases} 1 & i \geq j \\ \sum_{v_k \in G.adj[v_i]} P(k,j) & i < j \end{cases}$$

全部加起来即可获得图中所有路径的数量。算法流程如下：

(1) 计算 DAG 的拓扑序列 $v_1,\ldots,v_k$（对图进行 DFS 再根据 post 序逆序输出即可）

(2) 对每个点 v_i，根据上述公式依次计算 $P(i,j)$，其中 $j \in \{i,\ldots,k\}$。

(3) 输出所有的 $P(i,j)$ 的和。

第一步的时间复杂度为 $O(|V|+|E|)$，第二步的时间复杂度为 $O(|V|\cdot|E|)$，第三步的时间复杂度为 $O(|V|^2)$，因此总的时间复杂度为 $O(|V|\cdot|E|)$. □