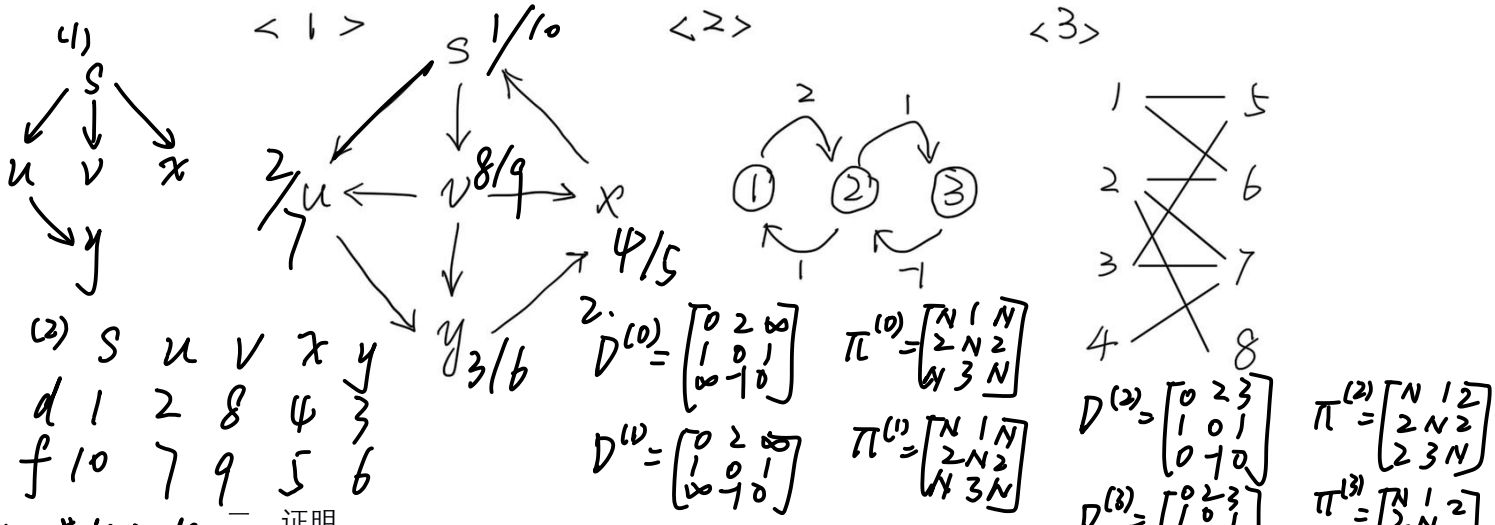
一：计算

1. （1）BFS 搜索树（2）DFS 搜索中，每个节点发现/结束时间；DFS 搜索树中的边种类
2. 求所有路径的最短路（写出距离矩阵与前驱矩阵）
3. 求最大二分匹配 → 3. 写过啦

1.



2. $D^{(0)}=\begin{bmatrix}0&2&\infty\\1&0&1\\\infty&7&0\end{bmatrix}$ $\pi^{(0)}=\begin{bmatrix}N&1&N\\2&N&2\\N&3&N\end{bmatrix}$

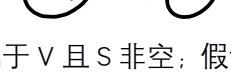
$D^{(1)}=\begin{bmatrix}0&2&\infty\\1&0&1\\\infty&7&0\end{bmatrix}$ $\pi^{(1)}=\begin{bmatrix}N&1&N\\2&N&2\\N&3&N\end{bmatrix}$

$D^{(2)}=\begin{bmatrix}0&2&3\\1&0&1\\0&7&0\end{bmatrix}$ $\pi^{(2)}=\begin{bmatrix}N&1&2\\2&N&2\\2&3&N\end{bmatrix}$

$D^{(3)}=\begin{bmatrix}0&2&3\\1&0&1\\0&7&0\end{bmatrix}$ $\pi^{(3)}=\begin{bmatrix}N&1&2\\2&N&2\\2&3&N\end{bmatrix}$

二、证明

4. C1、C2 为两个强连通分量，假设边(u,v)属于 E（u 属于 C1、v 属于 C2）；证明 f(C1)>f(C2)（f 表示 dfs 搜索结束时间）



4. ①若 d(C1)<d(C2)
设 x 是 C1 中最先被发现的点，则在 d[x] 时刻 x 到 G 中任意点均有白色路径，则 C1 中的点均为 x 后代；同理 C2 存在到 C2 的路径，则也为白色路径，则 C2 中的也为 x 后代，则 f(C1)=f[x]>f(C2)

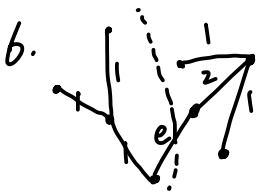
②若 d[C1]>d(C2)
设 y 是 C2 中最先被发现的点，则在 d[y] 时刻 y 到 C2 中任意点存在白色路径则 C2 中的点均为 y 后代，则 f(C2)=f(y)
同为存在 C1 到 C2 的路则必不存在 C2 到 C1 的路，则 C2 中无法到达 C1 中的就结束则 f(C2)=f(y)<f(C1)

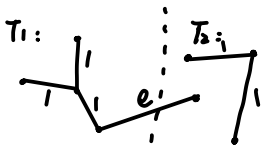
5. 对于点集 S，有 S 真属于 V 且 S 非空；假设存在权值最小的边 e(u,v)（u 属于 S，v 属于 V-S），证明最小生成树必包含边 e

5. 证：假设存在一棵 MST T' 不包含 e，则在该割上必存在一条边 e'(u',v'), u'∈S, v'∈S，则将 e'(u',v') 去掉，形成两个连通块，将 e(u,v) 加入形成树 T，则 T=T'-{e'}+{e}
又因为 w(e)<w(e') 则 w(T)<w(T') 则 T' 不是 MST，不符合假设

三、判断（正确给出证明，错误给出反例）

6. (X,Y)是图 G 的一个割，假设权值最小的边 e(u,v)（u 属于 X，v 属于 Y），T1 为 X 的最小生成树、T2 为 Y 的最小生成树，判断：T=T1+T2+{e}是 G 的最小生成树





e 是 (T1,T2) 的割边，若 T1 是最小生成树，则下来必是

7. d 表示距离，p 表示前驱，对一个图执行下图初始化过程（INIT）并进行任意次松弛（Relax）操作：

(1) 边 e(x,y)在经过 Relax(x,y,w)的瞬间，判断：d[y]≤d[x]+w(x,y)；

(2) 执行所有 Relax 后，对于 p[y]=x，判断：d[y]≤d[x]+w(x,y)；

<7>

```
INIT()
for: u ∈ V
    d[u] = +∞
    p[u] = N
d[s] = 0
```

```
Relax(x,y,w)
if: d[y] > d[x] + w(x,y)
then d[y] = d[x] + w(x,y)
     p[y] = x
```

7. (1) 是.

① 若 Relax(x,y,w) 前有 d[y]>d[x]+w(x,y)
Relax(x,y,w) 后瞬间有 d[y]=d[x]+w(x,y)

② 若 Relax(x,y,w) 前有 d[y]≤d[x]+w(x,y)
则 Relax(x,y,w) 后瞬间仍有 d[y]≤d[x]+w(x,y)

综上，……

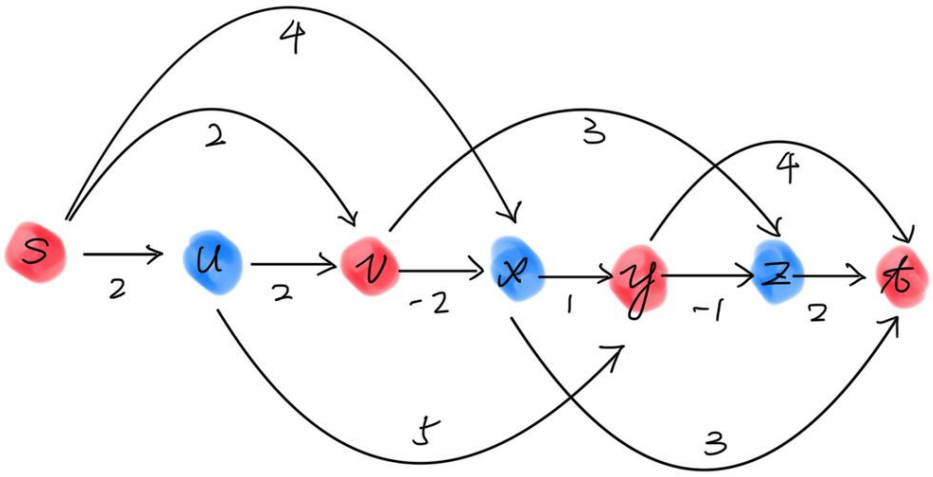
(2) 不正确，
因为松弛操作是任意的次序不能保证，
若 Relax(x,y,w) 后有 d[y]≤d[x]+w(x,y)
后又执行 Relax(z,x,w') 有 d[x]↓，则 d[y]≥d[x]+w(x,y)

四、程序设计题

8. 动态规划计算 s 到 t 的交替颜色最长路（即最长路上的节点必须依次为红蓝红蓝……）

（1）变量定义（2）递推式（3）计算下图

<8>



9. 设计算法计算 s 到其他所有点的最大路径容量（路径容量即路径上所有的容量最小值，最大路径容量即找到路径容量最大的路径）

（1）根据 Dijkstra 算法设计该算法

（2）计算下图每个点的最大容量、前驱节点

（3）证明算法正确性

d[v] 表示 S到 v 的最大容量

π[v] 表示最大容量路径上的前驱节点.

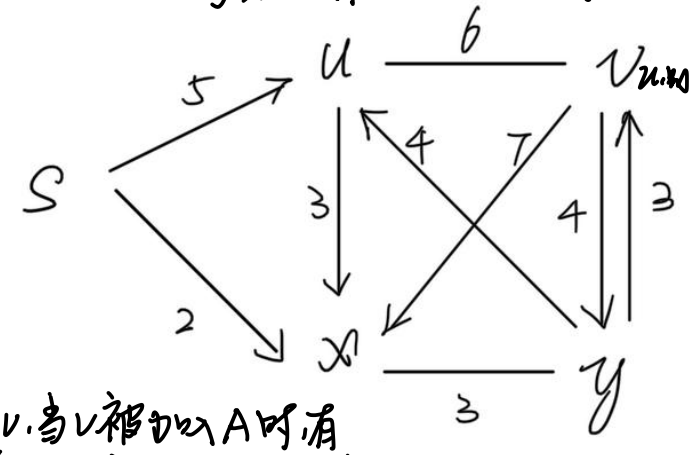
对于∀v∈V-s. d[v]初始化为 ∞. d[s]=0. 分为两个集合 A. B (B=V-A)

对∀v∈V-s, 若 d[v] 确定, 则将 v 放入 A.

每次从 B 取 d[v] 最大点, 将 v 放入 A, 并用 v 更新其在 B 中的邻接点的 d 值; 设含邻接点为 u. d[u]=max{d[u], min{d[v], w(v,u)}}

重复至 B 为空集

<9>



证明: 即证对于∀v. 当 v 被加入 A 时有

d[v]=δ(s,v), (δ(s,v) 表示 s 到 v 的最大容量)

证: 设存在 x∈V. 加入 A 时 d[x]<δ(s,x)

则 v 是其中 δ 最大的, v 一定不是 s 且是 s 可达的.

第一条: v 为 v 的最大容量路

在 v 加入前. 有 x∈A, y∈B. 则 d[x]=δ(s,x), d[y]=max{d[y], min(d[x], w(x,y))}

=min(d[x], w(x,y))=δ(s,y)

则 d[y]=δ(s,y) ≥ δ(s,v) > d[v]

则有 d[y]>d[v]

又因为 v 比 y 先加入 A. 则 d[v]≥d[y]

则 d[y]=d[v]

y=v

考试时间：2020 年 9 月 5 日

总的来说和 2019-2020 年的题几乎一样，19-20 年的题可参考 https://blog.csdn.net/weixin_43371116/article/details/104736487

一.
（1）强连通分量正确性证明。
（2）DAG 中最长路径的算法设计，写出 bellman 方程和伪代码，并分析时间复杂度。

二.
（1）白色路径定理的证明。
（2）假设最短路径含有 K 条边，证明迭代 K 次可以产生最短路径。

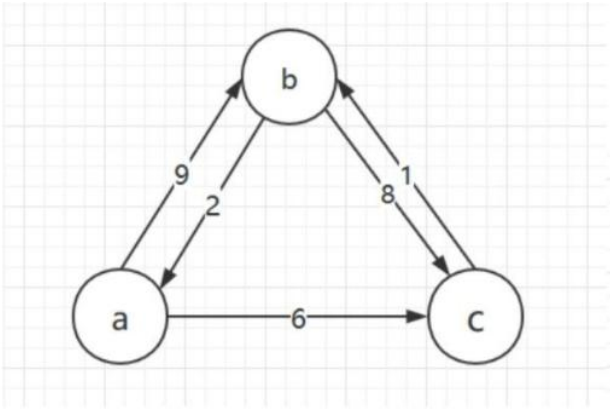
三.
T 是图 G 中的一棵最小生成树，现将 G 中一条边的权重改为 w'，设计算法实现对最小生成树 T 的更新。简述思想，写出伪代码，分析正确性。

四.
给了一个图，计算出最大流和最小割，要给出详细过程。

五.
每条边赋予一个宽度，一条路径的宽度为这条路径上边的最小宽度。借鉴 Dijkstra 算法思想，计算出从源点 S 到其他每个顶点的路径的最大宽度。简述基本思想，写出伪代码，证明正确性，分析最坏时间复杂度。

第三套：山大软院算法期末题回忆版
可能乱序 and 有差错，仅供参考
老师会捞的 题目都很简单 不需要太复习

1.三种时间复杂度比较异同
2.T(n)=T(n*3/4)+nlogn 求 T(n)的最大上界
3.npc 问题证明
4.强连通分量 算法思想和证明
5.图三个证明
（1）证明最短路的子路也是最短路
（2）不记得
（3）δ(s,v)<=δ(s,u)+w(u,v)
6.Floyd 算法 简述思想，时间复杂度，填最短路径填补矩阵 D
（图片仅供参考，好像是这个图）



$$\begin{Bmatrix} 0 & 2 & 8 \\ 2 & 0 & 6 \\ \infty & 1 & 0 \end{Bmatrix}$$

7. 已知图 G 和最小生成树 T，将 G 中一条边权值降低，求更新最小生成树。

8.动态规划算法 合并硬币堆最小耗费 耗费值是两个硬币堆数量之和

9.贪心算法 求最少点命中多个闭区间

计算题

- DFS：画出深度优先树；给出每个点的开始时间和结束时间；给出每条边的分类
- 有向图上的多源最短路径，要求计算$distance\ matrices\ and\ predecessor\ matrices$。
- 最大二分匹配

证明题

- 对于两个连通分量$C1\ C2$，存在边(u,v)，$u \in C1, v \in C2$。证明$f(C1) > f(C2)$
- 对于有向图G，各边权重不同。目前存在一划分$S, V-S, S \neq \emptyset$，边$e=(u,v), u \in S, v \in V-S$，且$e$是横跨划分权重最小的边。证明：任何一棵$MST$均包括$e$。

判断题

- 对于一连通图G，我们有一划分$S \quad V-S$，并且G中权重最小的边$e=(u,v)$，$u \in S, v \in V-S$，记$S \quad V-S$的生成子图分别为$X\ Y$，$X\ Y$的最小生成树分别记为$T1\ T2$，判断$T1 \cup T2 \cup e$是否是G的最小生成树。如果是，给出简短解释，否则举出反例。 (不确定表述是否和原题完全一致，大概意思如此)

- 给出初始化的操作和$RELAX(u,v,w)$的伪代码（与课本一致），之后进行一系列的松弛操作（原题并未明确说明具体顺序之类的详细信息，仅仅指明进行了松弛操作）

  - 在完成$RELAX(u,v,w)$的瞬间，$d[y] \leq d[x] + w(u,v)$
  - 在完成所有松弛操作之后，如果$y.\pi = x$，则$d[y] \leq d[x] + w(u,v)$

  以上两个命题，哪个是正确的，哪个是错误的？如果正确给出证明，否则举出反例。

  （注：一对一错，上面为正确，下面是错误的，关键在于题目未给出松弛操作的详细信息——不保证解答正确）

算法设计题

- 在有向图中每个节点都有颜色。或者为红色，或者为蓝色，设计一个DP算法找出s到t的**最长**红蓝交替路径。（红蓝交替路径即路径上节点颜色交替）

  要求：给出变量定义；给出变量的递推关系；在给出的实例上运行算法

- 有向图G中每条边的权重表示容量，记为$c(u,v)$。对于一条路径$p =< s, ..., t >$来说，其容量为路径上各边容量的最小者。对于除起点s之外的图中的每一点t，存在一个最大容量的路径，原题定义$\phi(t)$表示该值。

  要求：

  模仿$Dijkstra$算法，设计算法求出每个点的ϕ值

要求：

- 模仿$Dijkstra$算法，设计算法求出每个点的ϕ值
- 在给出的实例上运行你的算法。给出了一个表格，包括每个点的最后结果及其前驱节点。
- 证明你的算法。