算法作业 11

孟妍廷 2015202009

2017 年 12 月 13 日

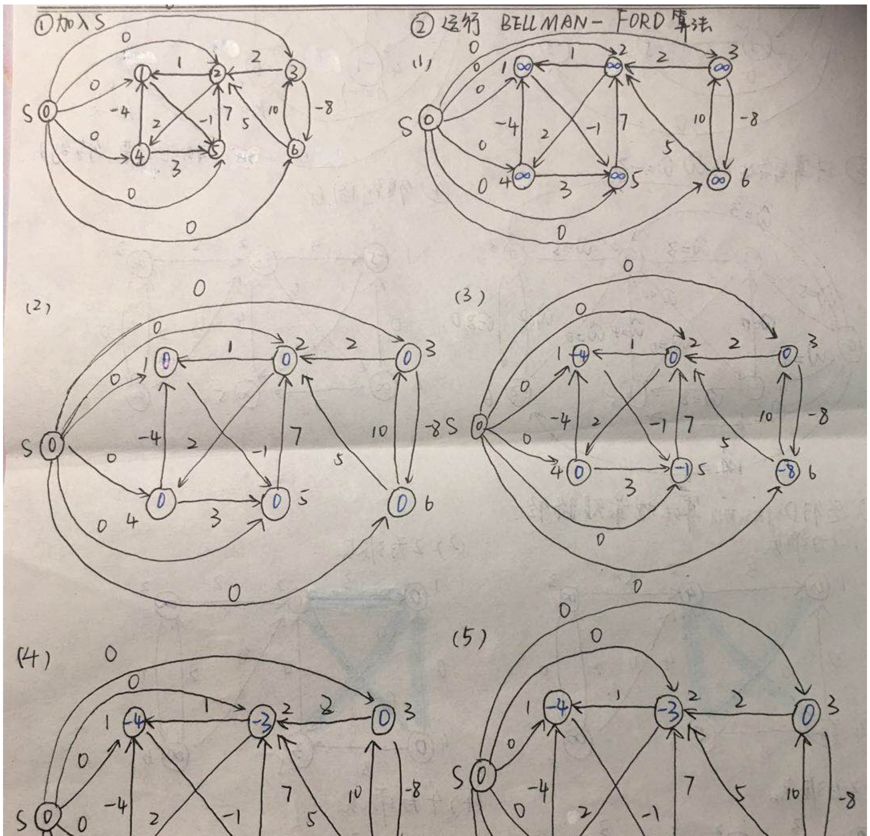
25.2-4

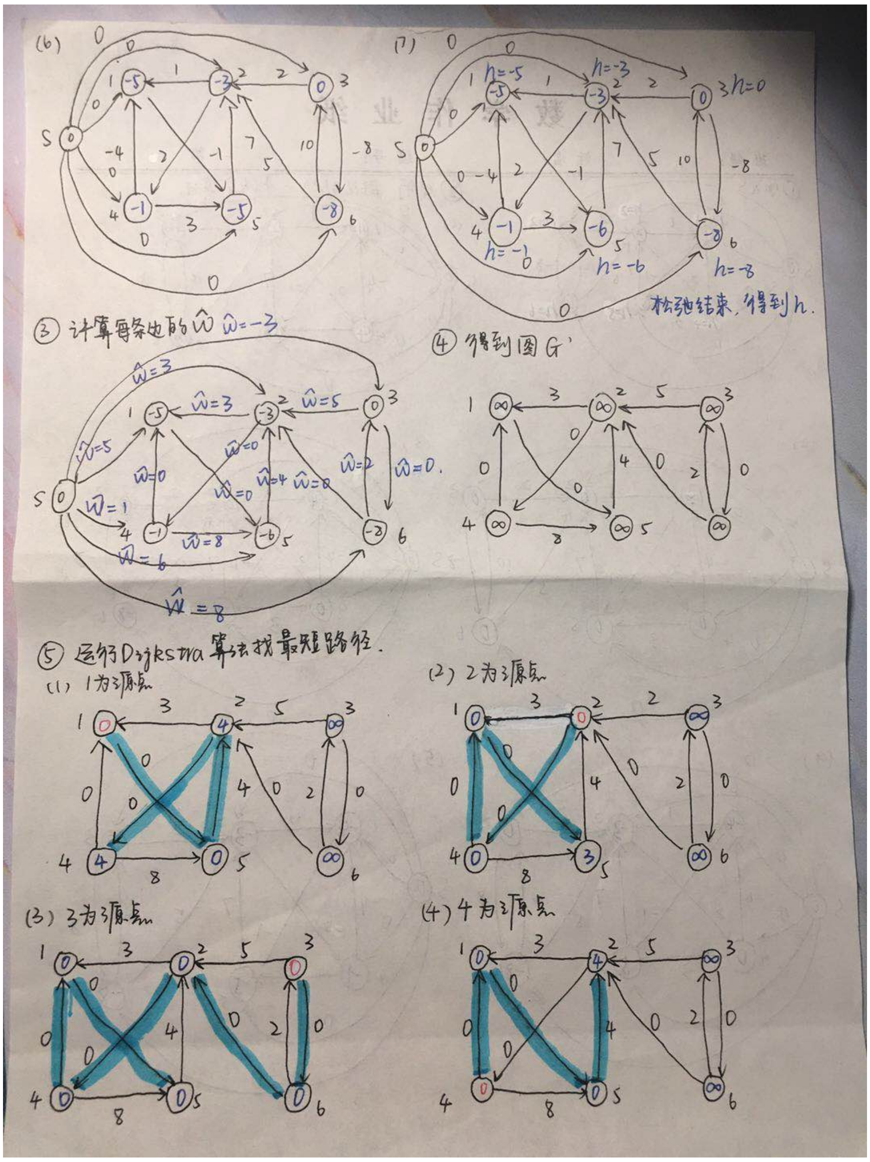
证明：在原始的 Floyed-Warshall 算法中，$d_{ij}^{(k)}$ 的意思是从结点 i 到结点 j 的所有中间结点全部取自集合 {1,2......k} 的一条最短路径的权重。每一个 $d_{ij}^{(k)}$ 都是在 $d_{ij}^{(k-1)}$ 的基础上算出来的，也就是说，在计算 $d_{ij}^{(k)}$ 时，是在包含之前在集合 {1,2......k-1} 中选出的结点的基础上判断是否加入结点 k，计算的过程占用了 $\Theta(n)$ 的计算时间。因此接下来要证明去掉所有上标的算法可以达到相同的目的。

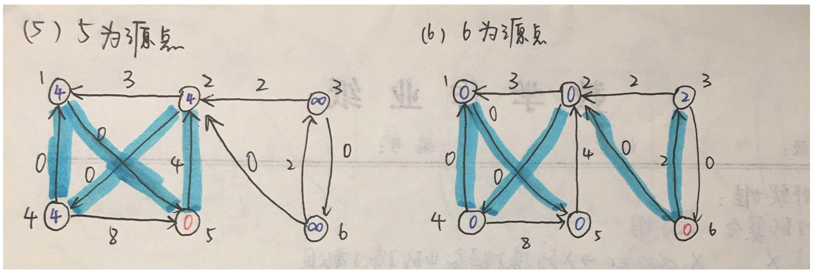
对于任意 $k \in \{1...n\}, d_{ij} = min(d_{ij}, d_{ik} + d_{kj})$ 的意思是判断结点 k 加入路径 (i,j) 中是否能使路径的权重变小，如果能，则把结点 k 加入路径 (i,j) 中，不能则保留原来的路径。则由于再循环中 k 是递增的，对于结点 k+1，判断 $d_{ij} = min(d_{ij}, d_{ik+1} + d_{k+1j})$ 时 d_{ij} 是已经包含了对结点 1...k 的判断。用书上的例子进项验证，最终二者得到的 D 矩阵是一样的。

因此得证，去除上标的算法是正确的。

25.3-1

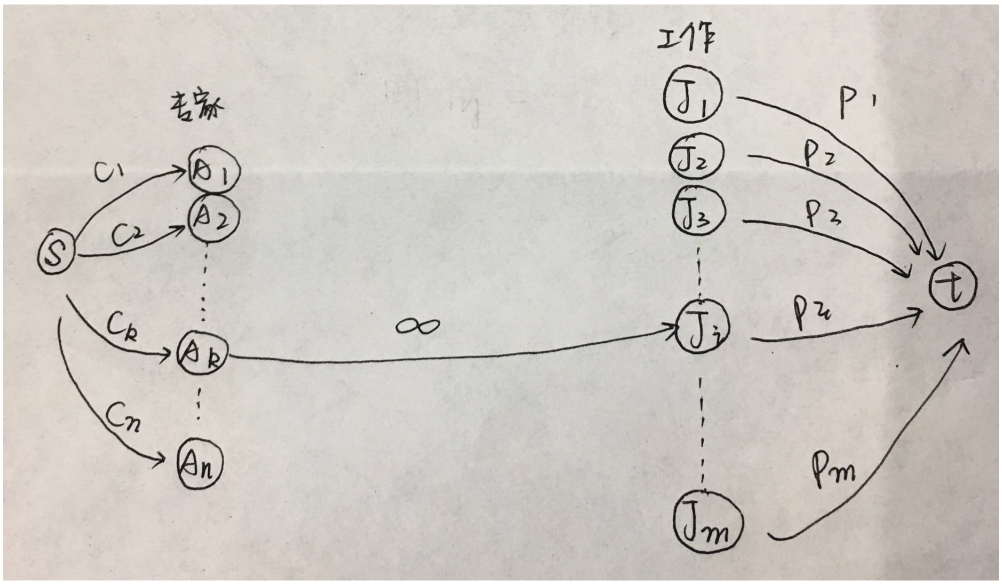






26-3

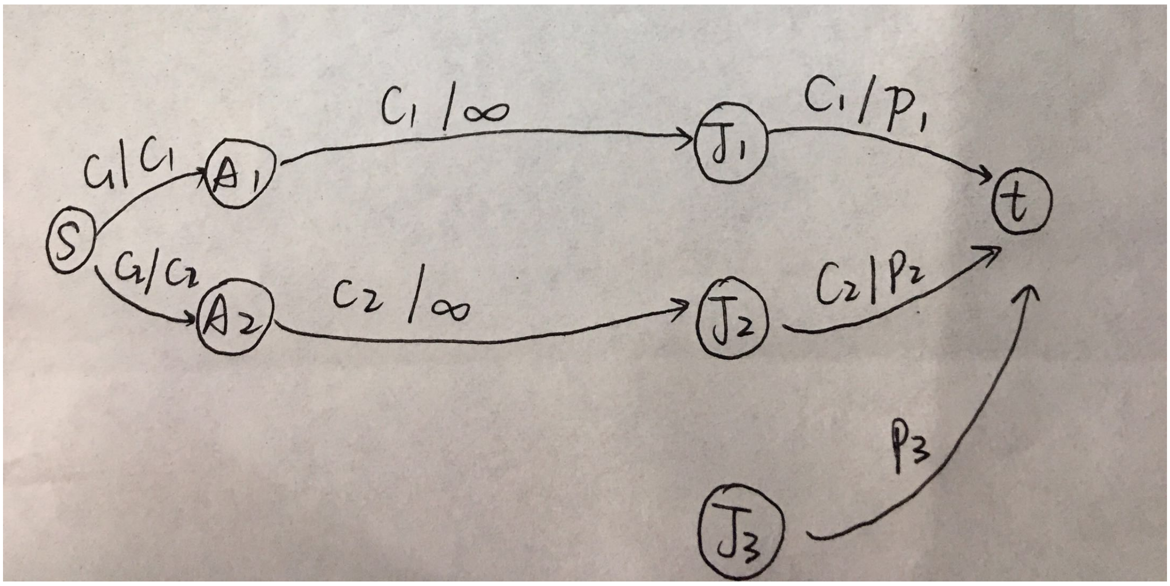
依题意可得，流图的形式如下：



a. 证明：由题目可知，对于结点 $A_k \in R_i$，则图 G 包含边 (A_k, J_i)，其容量为 ∞，又对于流图的切割 (S,T), 其容量是从集合 S 发出进入集合 T 的边的容量之和。

所以若 $J_i \in T, A_k \notin T$，则一定有一条由 A_k 指向 J_i，容量为 ∞ 的边流入 T，与切割是有限容量矛盾，所以一定有 $A_i \in T$

b. 解：以下图为例：



可知求最大流后指向 t 的容量为被选中的教授的聘请费用与未被选中的项目的营业额之和，即为 $\sum$ 被选中$c_k + \sum$ 未被选中p_i. 由最大流最小割定理可知，这是这是最小割的容量，即为最小容量。

又因为最大净收入 $= max(\sum$ 被选中$p_i - \sum$ 被选中$c_k) = max(\sum_{i=1}^{m} pi - \sum$ 未被选中$p_i - \sum$ 被选中$c_k) = \sum_{i=1}^{m} pi - min(\sum$ 被选中$c_k + \sum$ 未被选中$p_i)$, 因此公司的最大净收入就是用所有项目的营业额之和减去最小切割的容量和。

c. 解：算法如下：

```
solution(G, s, t)
  //用一个数组A[1...m]记录每一个领域聘请专家的费用
  FORD − FULKERSON(G, s, t) //对该流图求最大流
  for each J_i in G
    if (J_i, t).f < c_f(J_i, t)
      for k = 1 to n
```

$if\ A[k] = (J_i, t).f$

$print\ A_k$领域的教授被选中做J_i项目

时间复杂度为 $O((m+n+r)|f^*|)$