Efficient Algorithms
Summer term 2013

Institut für Theoretische Informatik
Leibniz Universität Hannover

Solution to exercise sheet 4

23.04.2013

Exercise 1: Given a directed graph $G = (V, E)$ with $V = \{1, 2, \dots, n\}$, we define the *transitive closure* of G as the graph $G^* = (V, E^*)$, where

$$E^* = \{ (i, j) \in V \times V \mid \text{there is a path from } i \text{ to } j \text{ in } G \} .$$

Usually Boolean values require less storage than words on current computers. Now construct an efficient algorithm to compute the transitive closure of a digraph which space requirement is less than the Floyd-Warshall algorithm.

Solution: The following algorithm uses only Boolean values rather than integer values, its space requirement is less than the Floyd-Warshall algorithm's by a factor corresponding to the size of a word of computer storage.
Define $t_k(i, j)$ to be 1 if there exists a path in G from i to j with all intermediate vertices in $\{1, \dots, k\}$ and 0 otherwise. We construct the transitive closure $G = (V, E^*)$ by putting edge (i, j) into E^* iff $t_n(i, j) = 1$:

$$t_0(i,j) = \begin{cases} 0 & \text{, if } i \neq j \text{ and } (i,j) \notin E, \\ 1 & \text{, if } i = j \text{ or } (i,j) \in E, \end{cases}$$

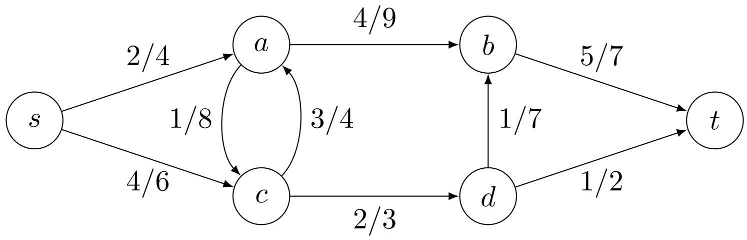
$$t_k(i,j) = t_{k-1}(i,j) \lor \big(t_{k-1}(i,k) \land t_{k-1}(k,j)\big), \qquad \text{if } k \geq 1.$$

Algorithm 1: Transitive-Closure($G = (V, E)$)

$n \leftarrow ||V||$;
for $i \leftarrow 1$ *to* n **do**
  for $j \leftarrow 1$ *to* n **do**
    if $i = j$ *or* $(i, j) \in E$ **then** $t_0(i, j) \leftarrow 1$;
    else $t_0(i, j) \leftarrow 0$;
for $k \leftarrow 1$ *to* n **do**
  for $i \leftarrow 1$ *to* n **do**
    for $j \leftarrow 1$ *to* n **do**
      $t_k(i,j) \leftarrow t_{k-1}(i,j) \lor \big(t_{k-1}(i,k) \land t_{k-1}(k,j)\big)$;

□

Exercise 2: Consider the following flow network with a given flow.



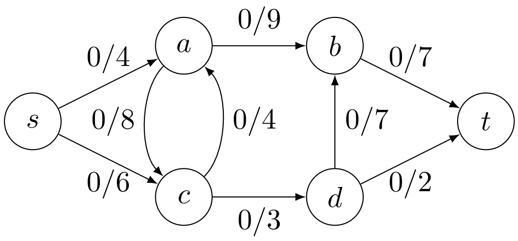
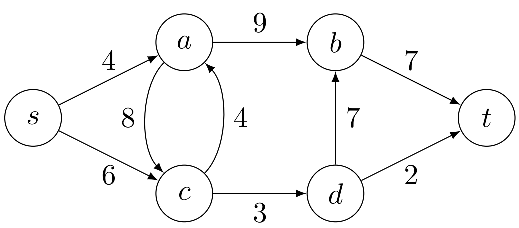
1. What is the flow across the cut $(\{s, c, d\}, \{a, b, t\})$? What is the capacity of this cut?

2. Show the execution of Ford-Fulkerson on the flow network.

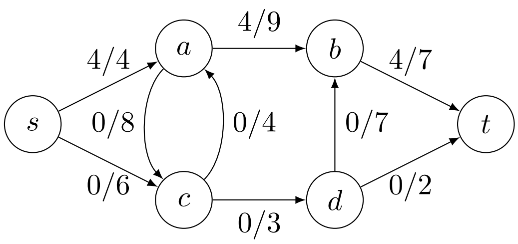
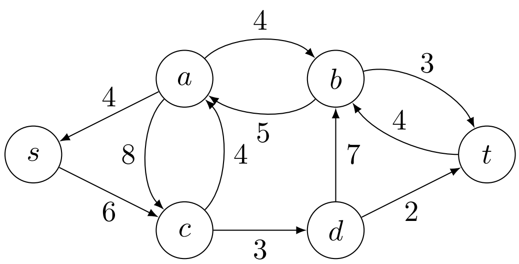
Solution:

1. $c(\{s, c, d\}, \{a, b, t\}) = 4 + 4 + 7 + 2 = 20$, and
   $f(\{s, c, d\}, \{a, b, t\}) = 2 + 3 + (-1) + 1 + 1 = 6$

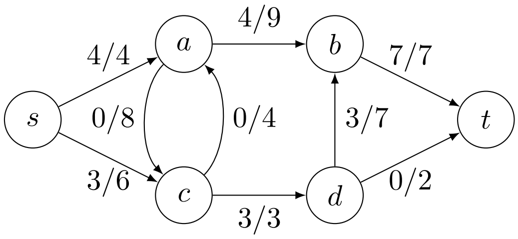
2. Execution of FF (left the current flow network with the flow, right the residual network):

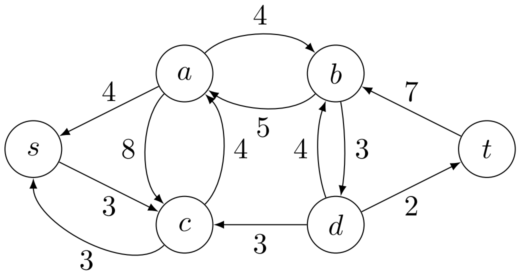



Consider the augmenting path $p_1 = s, a, b, t$.

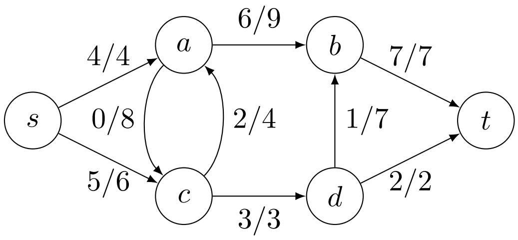



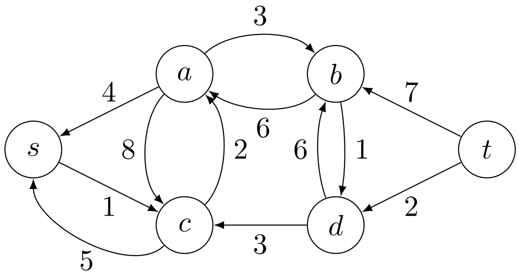
Augmenting path $p_2 = s, c, d, b, t$.





Augmenting path $p_3 = s, c, a, b, d, t$.





Now there does not exist any other augmenting path and the algorithm terminates.

$\square$