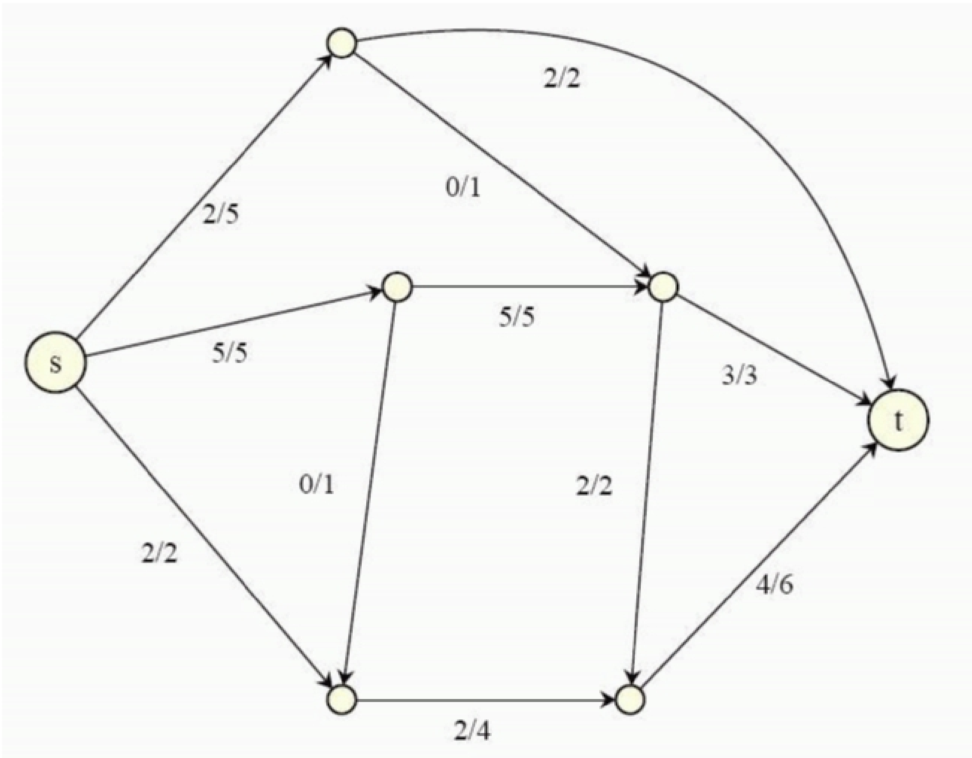
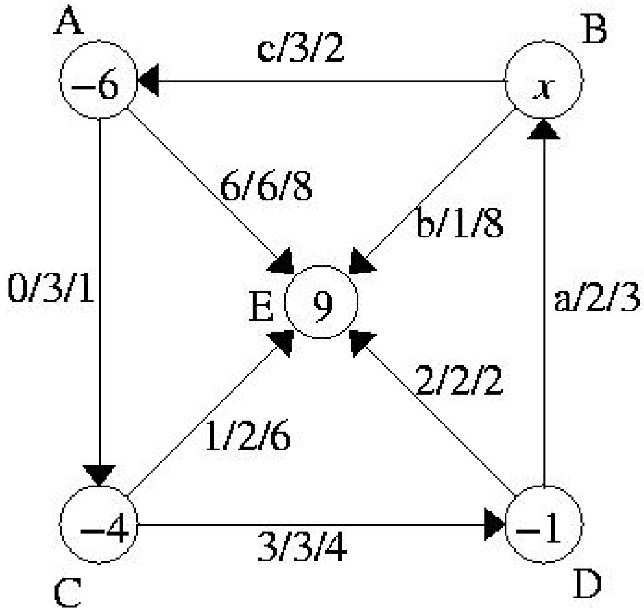
## Problem A (17%)




In the flow network illustrated, we are in the process of executing the Ford-Fulkerson algorithm and have already performed several augmentations. Each arc in the network is labeled with two numbers; the first being the amount of flow in our current solution and the second being the capacity of the arc. Finish the execution of the Ford-Fulkerson algorithm and make a similar figure indicating the maximum flow. Indicate also in your figure a minimum cut.

## Problem B (16%)

In the following picture a flow network $G$ with costs and a feasible flow $f$ in $G$ is given.




Each node is labeled with its balance. The three numbers stated along each arc are (in order) the *flow* along this arc, the *capacity* of this arc and the *cost* of this arc. For instance, $b_A = -6$, $f_{(C,E)} = 1$, $c_{(C,E)} = 2$ and $k_{(C,E)} = 6$. The balance of the node $B$ and flows on arcs adjacent to $B$ are not stated and have been replaced with symbols $x, a, b, c$. Other flows and costs that are not stated are as given by the skew symmetry rules, or they are zero.

1. For which values of $x, a, b, c$ is the stated flow a feasible flow?
2. Draw the residual network $G_f$ (with $x, a, b, c$ being those values found in question 1).
3. List all augmenting cycles in $G_f$ (indicate each cycle as a list of vertices).

## Problem C (17 %)

Consider the following five linear programming dictionaries:

Dictionary a:

$$\begin{array}{rcrcrcrcr} x_1 & = & 2 & - & 10x_3 & + & 10x_4 & - & x_6 \\ x_2 & = & & - & x_3 & - & 10x_4 & - & 2x_6 \\ x_5 & = & & - & 10x_3 & - & x_4 & - & 2x_6 \\ \hline z & = & 7 & - & x_3 & - & x_4 & - & 2x_6 \end{array}$$

Dictionary b:

$$\begin{array}{rcrcrcrcr} x_2 & = & & & 2x_1 & - & 3x_3 & - & x_4 \\ x_5 & = & & - & x_1 & - & 2x_3 & - & 2x_4 \\ x_6 & = & 8 & - & 3x_1 & - & 4x_3 & - & 2x_4 \\ \hline z & = & 1 & - & 2x_1 & - & 4x_3 & - & 8x_4 \end{array}$$

Dictionary c:

$$\begin{array}{rcrcrcrcr} x_2 & = & 10 & - & 2x_1 & - & 2x_3 & - & x_4 \\ x_5 & = & 1 & - & 4x_1 & - & x_3 & - & 2x_4 \\ x_6 & = & 20 & - & 3x_1 & - & 4x_3 & - & 2x_4 \\ \hline z & = & 3 & - & 2x_1 & + & x_3 & + & 2x_4 \end{array}$$

Dictionary d:

$$\begin{array}{rcrcrcrcr} x_1 & = & 1 & - & 2x_3 & - & x_4 & - & x_6 \\ x_2 & = & 2 & - & 4x_3 & - & 10x_4 & - & 2x_6 \\ x_5 & = & & - & 3x_3 & - & x_4 & - & 2x_6 \\ \hline z & = & 7 & - & x_3 & + & x_4 & + & 2x_6 \end{array}$$

Dictionary e:

$$\begin{array}{rcrcrcrcr} x_4 & = & 6 & - & 2x_1 & + & 3x_2 & - & x_3 \\ x_5 & = & 5 & - & 4x_1 & + & x_2 & - & 2x_3 \\ x_6 & = & 3 & - & 3x_1 & + & 2x_2 & - & 2x_3 \\ \hline z & = & 1 & + & 2x_1 & + & 4x_2 & + & x_3 \end{array}$$

Copy the table on the following page and fill it in as indicated below:

- In the row labeled “Feasible”, indicate for each dictionary if it is feasible. Write **Y** if it is, and write **N** if it is not.
- In the row labeled “Terminal”, indicate for each dictionary if it is terminal, i.e., if it is a final dictionary in the execution of the simplex algorithm *and* the corresponding basic solution is optimal. Write **Y** if it is, and write **N** if it is not. Write **N/A** if the present dictionary is not feasible.
- In the row labeled “Degenerate”, indicate for each dictionary if the corresponding basic solution is degenerate. Write **Y** if it is, and write **N** if it is not. Write **N/A** if the present dictionary is not feasible.

- In the row labeled "Entering", write for each dictionary the *name* of the variable who will enter the basis in the next iteration of the simplex algorithm, *assuming that Bland's rule is applied.* Write **N/A** if there is no such variable or if the present dictionary is not feasible.
- In the row labeled "Leaving", write for each dictionary the *name* of the variable that will *leave* the basis in the next iteration of the simplex algorithm, *assuming that Bland's rule is applied.* Write **N/A** if there is no such variable or if the present dictionary is not feasible.
- In the row labeled "New value", write the value of the objective function for the basic solution corresponding to the dictionary obtained *after* performing one pivot according to Bland's rule. Write **N/A** if there is no such next dictionary or if the present dictionary is not feasible.

| | a | b | c | d | e |
|---|---|---|---|---|---|
| Feasible | | | | | |
| Terminal | | | | | |
| Degenerate | | | | | |
| Entering | | | | | |
| Leaving | | | | | |
| New value | | | | | |

## Problem D (16%)

The E.R. of a major hospital receives 134 patients in need of emergency treatment. Each of the 134 patients requires a transfusion of one unit of whole blood. The E.R. has supplies of 135 units of whole blood. The number of units of blood available in each of the four major blood groups and the distribution of patients among the groups is summarized below.

| Blood type | A | B | O | AB |
|---|---|---|---|---|
| Supply | 46 | 34 | 45 | 10 |
| Demand | 39 | 38 | 42 | 15 |

Type A patients can only receive type A or O; type B patients can receive only type B or O; type O patients can receive only type O; and type AB patients can receive any of the four types.

Formulate the problem of determining a distribution that satisfies the demands of a maximum number of patients as a max flow problem. Draw the entire flow network as a directed graph and put the arc capacity above each arc. You do *not* have to solve the resulting max flow problem.

## Problem E (17%)

A cargo plane has three compartments for storing cargo: front, centre and rear. These compartments have the following limits on both weight and space:

| Compartment | Weight capacity (tons) | Space capacity (cubic metres) |
|---:|---:|---:|
| Front | 10 | 6800 |
| Center | 16 | 8700 |
| Rear | 8 | 5300 |

Furthermore, the weight of the cargo in the respective compartments must be the same proportion of that compartment's weight capacity to maintain the balance of the plane. For instance, if we put 8 tons in the center compartment, we have to put 5 tons in the front compartment, since $8/16 = 5/10$.

The following four cargoes are available for shipment on the next flight:

| Cargo | Weight (tons) | Volume (cubic metres/ton) | Profit (DKK/ton) |
|---:|---:|---:|---:|
| C1 | 18 | 480 | 310 |
| C2 | 15 | 650 | 380 |
| C3 | 23 | 580 | 350 |
| C4 | 12 | 390 | 28 |

Any proportion of these cargoes can be accepted. The objective is to determine how much (if any) of each cargo C1, C2, C3 and C4 should be accepted and how to distribute each among the compartments so that the total profit for the flight is maximised.

Show how to express this problem as a linear program. You do *not* have to solve the resulting program.

## Problem F (17 %)

Let $U$ be a finite set. Given a finite family $S_1, S_2, \ldots, S_k$ of non-empty finite subsets of $U$, we want to find a subfamily $S_{i_1}, S_{i_2}, \ldots, S_{i_k}$ where $\{i_1, i_2, \ldots, i_k\} \subseteq \{1, 2, \ldots, k\}$ and so that all sets in the subfamily are *disjoint*. We refer to such a subfamily as a *disjoint subfamily*. We want to find as large a subfamily as possible; i.e., we want to maximize $k$.

As an example, let $U = \{1, 2, .., 10\}$ and let the family be $S_1 = \{1, 2, 5\}$, $S_2 = \{1, 2, 7\}$, $S_3 = \{2, 7, 8, 10\}$, $S_4 = \{1, 3\}$. Here the largest possible subfamily is $\{S_3, S_4\}$, i.e., we can have $k = 2$. Indeed, it is not possible to have a disjoit subfamily of size 3, as the element 2 is contained in all sets, except the set $S_4$.

Given a family, show how to express as an integer linear program the problem of finding a largest possible disjoint subfamily for the family. Explicitly state what your program would look like for the stated example and what the optimum solution (as an integer vector) for this example would be.