NOTE TO STUDENTS: This file contains sample solutions to the term test together with the marking scheme for each question. Please read the solutions and the marking schemes carefully. Make sure that you understand why the solutions given here are correct, that you understand the mistakes that you made (if any), and that you understand *why* your mistakes were mistakes.

Remember that although you may not agree completely with the marking scheme given here it was followed the same way for all students. We will remark your test only if you clearly demonstrate that the marking scheme was not followed correctly.

For all remarking requests, please submit your request **in writing** directly to your instructor. For all other questions, please don't hesitate to ask your instructor during office hours or by e-mail.

Question 1. [5 MARKS]

Let N be any flow network with integer capacities and max flow value F. Let p be some simple path from source s to sink t in N. Let N' be the network obtained from N by adding 1 to the capacity of every edge on the path p.

For each of the claims below, decide whether it is true or false. If true, give a brief justification, if false, give a counterexample.

Part (a) [3 MARKS]

The max flow of N' is *at least* $F + 1$.

SAMPLE SOLUTION:

> The statement is **true**. Let f be a max flow in N ($|f| = F$), and denote its residual network by N_f. Then f is also a flow in N', and its residual network N'_f is obtained from N_f by adding 1 to the capacities of all edges along the path p. Hence, p is an augmenting path of residual capacity 1 in N'_f. So, the max flow value in N' is at least $|f| + 1 = F + 1$.

MARKING SCHEME:

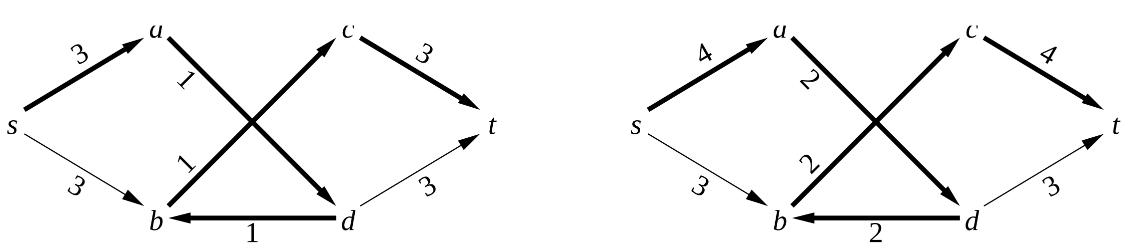
A. 1 mark: correct answer
B. 2 marks: correct justification

Part (b) [2 MARKS]

The max flow of N' is *exactly* $F + 1$.

SAMPLE SOLUTION:

> The statement it **false**. It is possible for p to cross the min cut more than once, and then the max flow could increase by more than 1. See counterexample.



> Before we increase the capacities, the max flow through the network on the left is 2. After we increase the capacities along $p = s \to a \to d \to b \to c \to t$, the max flow through the network becomes $4 > 2 + 1$.

Marking Scheme:

A. 1 mark: correct answer
B. 1 mark: correct counter-example

Question 2. [12 marks]

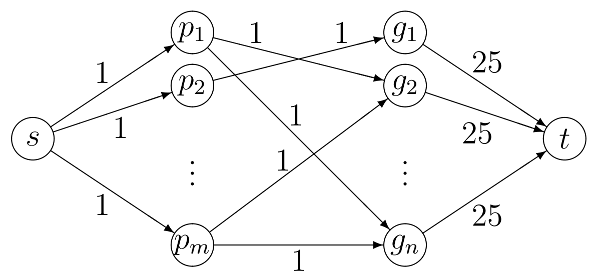
A botanical garden is planning to build n greenhouses, each of which is intended to represent a different climate zone. They have a list of m kinds of plants that they would like to grow there. Not every kind of plant can grow in every greenhouse, though usually one kind can grow in more than one climate; so associated with every kind of plant i is a list C_i of greenhouses where it can grow. In addition to that, every greenhouse can host no more than 25 varieties of plants.

Given the n, m and C_i for $1 \leqslant i \leqslant m$, we would like to know the maximal number of different varieties of plants that can be grown in these greenhouses (total over all greenhouses).

Model this problem as a network flow problem (don't forget to specify clearly all edge directions and capacities in your network), and justify the correctness of your network. Give an efficient algorithm to solve this problem, and analyze the worst-case running time of your algorithm.

SAMPLE SOLUTION:

Modeling: Create a flow network $N = (V, E)$ as pictured below, where node p_i denotes the i^{th} plant variety and node g_j denotes the j^{th} greenhouse. There is an edge (p_i, g_j) if and only if $g_j \in C_i$. The capacities of edges (s, p_i) and (p_i, g_j) are all 1 and the capacities of edges (g_j, t) are all 25.



Justification: Every valid assignment of plant varieties to greenhouses yields a valid flow in N, by setting $f(s, p_i) = 1$ if p_i has been assigned to some greenhouse, $f(s, p_i) = 0$ otherwise; $f(p_i, g_j) = 1$ if p_i has been assigned to greenhouse g_j, $f(p_i, g_j) = 0$ otherwise; and $f(g_j, t) =$ the number of plants assigned to greenhouse j. Such a flow respects both the capacity and conservation constraints.

Conversely, every valid integer flow in N yields a valid assignment of plant varieties to greenhouses, by assigning the i^{th} plant variety p_i to the j^{th} greenhouse g_j such that $f(p_i, g_j) = 1$. Because the capacity into p_i is exactly 1, each plant variety can be assigned to at most one greenhouse. Because the capacity out of g_j is 25, each greenhouse j is assigned at most 25 new plant varieties.

Hence, the value of a maximum flow is equal to the maximum number of plant varieties that can be assigned to greenhouses and the algorithm is guaranteed to produce an optimal solution.

Algorithm:

MaximalAssignment(N):
  find a maximal flow f in N
  assign plant variety p_i to greenhouse g_j if and only if $f(p_i, g_j) = 1$.

Complexity: Using Edmonds-Karp to find a maximum flow, we obtain a complexity of

$$O(|V||E|^2) = O((m + n + 2)(m + mn + n)^2) = O((m + n)(mn)^2) = O(m^3n^2 + m^2n^3).$$

MARKING SCHEME:

- 4 marks: clear attempt to give a network-based algorithm to solve the problem (constructing a network from the problem input, solving the maximum flow problem on this network, and extracting a solution from the network flow), to argue that the algorithm is correct, and to analyze the algorithm's running time
- 4 marks: correct network definition (nodes, edges and capacities), and correct justification that network flow values and assignments correspond to each other (in both directions)
- 4 marks: correct algorithm, correct assignments constructed from a maximum flow, and correct analysis of algorithm's runtime (in terms of the number of inputs m and n).

Question 3. [8 marks]

Maximum Bipartite Matching Problem: Given an undirected bipartite graph $G = (V_1, V_2, E)$, where $E \subseteq V_1 \times V_2$, find a set of disjoint edges of the maximum size (where two edges are *disjoint* if they have no common endpoint).

Give a linear (or integer) program that corresponds to this problem. Describe clearly every component of your answer. Then justify the correctness of your linear program: explain clearly what each variable and constraint represents and how solutions to each problem correspond to each other (and what that tells you about the relative values of those solutions).

Sample Solutions:

Let $V_1 = \{u_1, \dots, u_k\}$ and $V_2 = \{w_1, \dots, w_\ell\}$.

Variables: $x_{i,j}$ for all i, j with $(u_i, w_j) \in E$ — *intention: matching* $M = \big\{(u_i, w_j) : x_{i,j} = 1\big\}$

Objective Function: maximize $\sum_{(u_i,w_j)\in E} x_{i,j}$ — *equal to* $|M|$

Constraints: $x_{i,j} \in \{0,1\}$, for all $(u_i, w_j) \in E$

$$\left.\begin{array}{l} \sum_{w_j\in V_2:(u_i,w_j)\in E} x_{i,j} \leqslant 1, \text{ for each } u_i \in V_1 \\ \sum_{u_i\in V_1:(u_i,w_j)\in E} x_{i,j} \leqslant 1, \text{ for each } w_j \in V_2 \end{array}\right\} \textit{no two edges in } M \textit{ have a common endpoint}$$

Every matching M over G gives rise to a feasible solution by setting $x_{i,j} = 1$ iff $(u_i, w_j) \in M$: since no two edges share an endpoint, every constraint is satisfied. Moreover, the size of M is equal to the value of the objective function, so the maximum value of the objective function is at least as large as the size of the maximum matching.

Every feasible solution to the integer program yields a matching M by selecting every edge (u_i, w_j) such that $x_{i,j} = 1$: the constraints guarantee that no two edges share an endpoint and the value of the objective is equal to the size of M. So, the size of the maximum matching is at least as large as the maximum value of the objective function.

Alternative Solution:

Variables: $x_{i,j}$ for all $1 \leqslant i \leqslant k = |V_1|$, $1 \leqslant j \leqslant \ell = |V_2|$

Objective Function: maximize $\sum_{i=1}^{k} \sum_{j=1}^{\ell} x_{i,j}$

Constraints: $x_{i,j} \in \{0,1\}$ for all i, j

$x_{i,j} = 0$ for all i, j such that $(u_i, w_j) \notin E$

$\sum_{j=1}^{\ell} x_{i,j} \leqslant 1$ for $i = 1, \dots, k$

$\sum_{i=1}^{k} x_{i,j} \leqslant 1$ for $j = 1, \dots, \ell$

Marking Scheme:

A. 2 marks: clear and correct definition of variables, with correct description of their meaning
B. 2 marks: clear and correct definition of the objective function, with correct description of its meaning
C. 2 marks: constraints clearly attempt to capture the property that no two edges share an endpoint, do so correctly, and this is stated clearly
D. 2 marks: reasonable explanation of the correspondence between the value of the objective function and the size of the matching (at most 1 mark if constraints are incorrect)