MATH1005/MATH6005 Discrete Mathematical Models
Final Exam, Semester 1, 2021

Photo Credit: Amithi Aparekka Liyanagamage.

Throughout this exam, we write $\mathbb{N}$ for the set of positive integers and $\mathbb{N}^\star$ for the set of non-negative integers; that is, $\mathbb{N} = \{1, 2, 3, \dots\}$ and $\mathbb{N}^\star = \{0, 1, 2, \dots\}$.

Problem 1 (10 marks) (a) Give an example of a statement that has the logical structure of an implication. Then write down the contrapositive, the converse and the inverse of your statement. Clearly label which statement is which (original, contrapositive, converse, inverse).

Original: If today is Monday, then tomorrow is Tuesday.

Contrapositive: If tomorrow is not Tuesday, then today is not Monday.

Converse: If tomorrow is Tuesday, then today is Monday.

Inverse: If today is not Monday, then tomorrow is not Tuesday

(b) Suppose that A and B are non-empty sets and that $R \subseteq A \times B$. What else must be true about R before we can say that R is an injective function?

(1) $\forall a \in A \ \exists!\, b \in B \ (a,b) \in R$

(if (1) is true, we write $R(a) = b$ when $(a,b) \in R$)

(2) $\forall a_1, a_2 \in A \ (R(a_1) = R(a_2) \Rightarrow a_1 = a_2)$

or $\forall a_1, a_2 \in A \ (a_1 \neq a_2 \Rightarrow R(a_1) \neq R(a_2))$

(c) Let P denote the set of prime numbers. Let s denote the following statement:

$$\exists k \in \mathbb{N} \ \ \forall n \in \mathbb{N} \ \ \big((2^n - 1 \in P) \to (n \leq k)\big)$$

The statement $\neg s$ is a famous conjecture in Number Theory. Write down a statement that is logically equivalent to $\neg s$ and in which the symbol $\neg$ does not appear.

$\neg s \equiv \neg \left[\exists k \in \mathbb{N} \ \forall n \in \mathbb{N} \ ((2^n - 1 \in P) \to n \leq k) \right]$

$\equiv \forall k \in \mathbb{N} \ \exists n \in \mathbb{N} \ \neg((2^n - 1 \in P) \to (n \leq k))$

$\equiv \forall k \in \mathbb{N} \ \exists n \in \mathbb{N} \ (2^n - 1 \in P) \wedge \neg(n \leq k)$

$\equiv \forall k \in \mathbb{N} \ \exists n \in \mathbb{N} \ (2^n - 1 \in P) \wedge (n > k)$

(d) Let $B = \{0,1\}^8$; for convenience we shall write $b_1 b_2 \ldots b_8$ as shorthand for $(b_1, b_2, \ldots, b_8)$. Let $\eta : B \to \mathbb{Z}$ be defined by the following rule

$$\forall b_1 b_2 \ldots b_n \in B \;\; \eta(b_1 b_2 \ldots b_n) = (-1)^{b_1} \times \left(b_2 \times 2^6 + b_3 \times 2^5 + b_4 \times 2^4 + b_5 \times 2^3 + b_6 \times 2^2 + b_7 \times 2^1 + b_8 \times 2^0\right).$$

Let $\tau : B \to \mathbb{Z}$ be the function that maps each element of B to the integer that it represents using the 8-bit signed integer method (also known as the "two's complement method", and the "toggle-plus-one" method).

(i) Evaluate $\tau(10100011)$.

$$\tau(10100011) = -(01011100 + 1)_2$$
$$= -(01011101)_2$$
$$= -(64 + 16 + 8 + 4 + 1)_{10}$$
$$= -(93)_{10}$$

(ii) Use set-roster notation to describe the range of η and the range of τ.

Since $2^7 = 128$, $\eta(11111111) = -(2^7 - 1) = -127$;
$\eta(01111111) = (2^7 - 1) = 127$.

So range $\eta = \{-127, -126, -125, \ldots, 127\}$

Since $2^7 = 128$, $\tau(10000000) = -(01111111 + 1)_2 = -(10000000)_2$
$= -128$

and $\tau(01111111) = 2^7 - 1 = 127$.

So range $\tau = \{-128, -127, -126, \ldots, 127\}$

(iii) Give two reasons why τ may be preferred to η as a method for representing integers in a computer.

(e.g. 2 of)

- Since η is not injective ($\eta(10000000) = \eta(00000000) = 0$) but τ is, one more integer can be represented with the same # of bits
- Algorithms don't have to deal with two different representations of 0 if we use τ instead of η.
- τ is better for implementing arithmetic in circuits.

Problem 2 (10 marks) (a) We define a sequence $(a_n)_{n\in\mathbb{N}}$ by

$$\begin{cases} a_1 = 1 \\ \forall n \in \mathbb{N}\ a_{n+1} = a_n \left(1 - \dfrac{1}{(n+1)^2}\right). \end{cases}$$

Use mathematical induction to prove that

$$\forall n \in \mathbb{N}\ \ a_n = \frac{n+1}{2n}.$$

Let $p(n)$: $a_n = \frac{n+1}{2n}$.

Basis step: LHS of $p(1) = a_1 = 1$ (by defn of the sequence)

RHS of $p(1) = a_1 = \frac{1+1}{2(1)} = \frac{2}{2} = 1$

Hence $p(1)$ is true.

Inductive step: Let $n \in \mathbb{N}$. Suppose that $p(1), p(2), \ldots, p(n)$ all hold. We consider $p(n+1)$.

LHS of $p(n+1) = a_{n+1}$

$= a_n\left(1 - \frac{1}{(n+1)^2}\right)$ (by defn of the sequence)

$= \frac{n+1}{2n}\left(1 - \frac{1}{(n+1)^2}\right)$ (by $p(n)$)

$= \frac{n+1}{2n} - \frac{n+1}{2n(n+1)^2}$

$= \frac{n+1}{2n} - \frac{1}{2n(n+1)}$

$= \frac{(n+1)^2 - 1}{2n(n+1)}$

$= \frac{n^2+2n+\cancel{1}-\cancel{1}}{2n(n+1)}$

$= \frac{\cancel{n}(n+2)}{2\cancel{n}(n+1)}$

$= \frac{n+2}{2(n+1)}$

$= \frac{(n+1)+1}{2(n+1)}$

$=$ RHS of $p(n+1)$

Hence $p(n+1)$ is true.

By the Principle of Mathematical Induction,

$\forall n \in \mathbb{N}\ p(n)$!

$\square$

(b) Use the logical equivalences below and the definitions of set operations to prove, or provide a counterexample to disprove, each of the following statements.

(i) For any universal set U and for any $A, B, C \in \mathcal{P}(U)$, we have

$$A \cap (B \cup C^c) = (A \cap B) \cup (A \setminus C).$$

Let U be a universal set. Let $A, B, C \in P(U)$.

Let $x \in U$. Then $x \in A \cap (B \cup C^c)$

$\iff x \in A \wedge x \in B \cup C^c$ (defn $\cap$)

$\iff x \in A \wedge (x \in B \vee x \in C^c)$ (defn $\cup$)

$\iff x \in A \wedge (x \in B \vee x \notin C)$ (defn ^{c})

$\iff (x \in A \wedge x \in B) \vee (x \in A \wedge x \notin C)$ (using 3. Distributivity below)

$\iff (x \in A \cap B) \vee (x \in A \setminus C)$ (defn $\cap$ and defn $\setminus$)

$\iff x \in (A \cap B) \cup (A \setminus C)$ (defn $\cup$)

(ii) For any universal set U and for any $A, B, C, D \in \mathcal{P}(U)$, we have:

$$\text{If } C \subseteq A \text{ and } D \subseteq B, \text{ then } (A \times B) \setminus (C \times D) = (A \setminus C) \times (B \setminus D).$$

The statement is false. eg. Let $U = \{a, b, 1, 2\}$,

$A = \{a, b\}$, $B = \{1, 2\}$, $C = \{a\}$ and $D = \{1\}$

Then $C \subseteq A$ and $D \subseteq B$ and

$(A \times B) \setminus (C \times D) = \{(a,1), (a,2), (b,1), (b,2)\} \setminus \{(a,1)\}$

$= \{(a,2), (b,1), (b,2)\}$ and

$(A \setminus C) \times (B \setminus D) = \{b\} \times \{2\} = \{(b,2)\}$..

Given any statement variables p, q, and r, a tautology t and a contradiction c, the following logical equivalences hold.

1. Commutative laws:	$p \wedge q \equiv q \wedge p$	$p \vee q \equiv q \vee p$
2. Associative laws:	$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	$(p \vee q) \vee r \equiv p \vee (q \vee r)$
3. Distributive laws:	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
4. Identity laws:	$p \wedge t \equiv p$	$p \vee c \equiv p$
5. Negation laws:	$p \vee \neg p \equiv t$	$p \wedge \neg p \equiv c$
6. Double negative law:	$\neg(\neg p) \equiv p$	
7. Idempotent laws:	$p \wedge p \equiv p$	$p \vee p \equiv p$
8. Universal bound laws:	$p \vee t \equiv t$	$p \wedge c \equiv c$
9. De Morgan's laws:	$\neg(p \wedge q) \equiv \neg p \vee \neg q$	$\neg(p \vee q) \equiv \neg p \wedge \neg q$
10. Absorption laws:	$p \vee (p \wedge q) \equiv p$	$p \wedge (p \vee q) \equiv p$
11. Negations of t and c:	$\neg t \equiv c$	$\neg c \equiv t$

Problem 3 (10 marks) (a) You and your friend are playing a board game. Each turn involves rolling three six-sided dice. The faces of each die are numbered 1, 2, 3, 4, 5, 6. Your friend, who is very reasonable but has not taken MATH1005/MATH6005, makes the following remark

> "These dice may be unfair. With three dice we can roll 16 different totals, so each total should appear every 16-th turn or so. However, we have been playing for hours and the total 18 has hardly ever come up."

In no more than five sentences, respond to your friend's statement. An excellent response will either agree or disagree with the reasoning in the statement, and will justify the position taken so clearly that your friend is likely to agree with you.

Your reasoning is incorrect. If we were to label the dice as A, B, C we can represent a roll by writing down 3 digits in order – the number on die A, the number on die B and the number on die C. There are $6^3 = 216$ different such sequences we can write and they are equally likely outcomes. Since only <u>one</u> of these outcomes gives a total of 18, we expect to see 18 as a total once every 216 turns or so.

(b) A **PIN** is a string of 4 digits. A PIN is said to be **non-repeating** if no digit appears twice. For example, 3137 is a PIN, while 0216 and 7935 are non-repeating PINs.

What is the probability that, when a PIN is selected at random, it is a non-repeating PIN in which the digits appear in strictly increasing order?

The experiment is to select a PIN at random. An outcome is a string of four digits. Let S denote the set of all such strings. Then

$|S| = 10 \times 10 \times 10 \times 10$ (# ways to choose the first digit
× # ——— 2nd —
× # ——— 3rd —
× # ——— 4th —)

Let E denote the set of four digits strings in which the digits are non-repeating and appear in strictly increasing order. Selecting such a string is equivalent to selecting a set of size 4 from the set $\{0,1,2,\dots,9\}$ because once you know which digit you have, they must be written down in strictly increasing order.. Thus $|E| = \binom{10}{4}$.

Since a PIN is selected at random in our experiment, the outcomes are equally likely. Thus

$$P(E) = \frac{|E|}{|S|} = \frac{\binom{10}{4}}{10^4} = \frac{\cancel{10}\times \overset{3}{\cancel{9}}\times \cancel{8}\times 7}{\cancel{4}\times\cancel{3}\times\cancel{2}\times 1} \frac{}{\cancel{10}\times 10\times 10\times 10} = \frac{21}{1000}$$

$$= 0.021$$

(c) For any $k \in \mathbb{N}$, an *XY-path* of length k in the Euclidean plane is a sequence of points

$$\big((x_n, y_n)\big)_{n\in\{0,1,2,\dots,k\}} \subseteq \mathbb{Z}\times\mathbb{Z}$$

such that

$$\forall n \in \{0,1,2,\dots,k-1\}\ \Big((x_{n+1}, y_{n+1}) = (1+x_n, y_n)\Big) \vee \Big((x_{n+1}, y_{n+1}) = (x_n, 1+y_n)\Big);$$

we say that such a path *starts at* (x_0, y_0) and *ends at* (x_k, y_k).

From all of the XY-paths that start at $(0,0)$ and end at $(4,6)$, one is chosen at random. What is the probability that the chosen XY-path visits the point $(2,0)$? Give your answer as a decimal, correct to two decimal places.

The experiment is to select an XY path that starts at $(0,0)$ and ends at $(4,6)$. Each such path must involve 4 moves in the x-direction and 6 in the y direction; so we represent each XY path by a string of 4 X's and 6 Y's in some order. Let S be the set of all such strings. Then $|S| = \binom{10}{4}$ (choose the 4 positions in which X's appear).

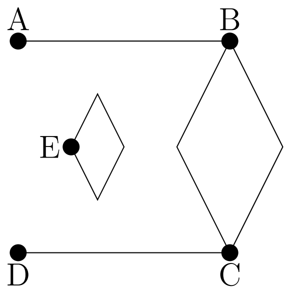
Let E be the event that the path visits $(2,0)$. We note that this only happens when the first 2 letters are X's. So $|E| = \binom{8}{2}$ (choose the 2 positions in which the last 2 X's appear)

Since the XY path is chosen at random, the outcomes are equally likely and

$$P(E) = \frac{|E|}{|S|} = \binom{8}{2}\Big/\binom{10}{4} = \frac{\cancel{8}\times\cancel{7}}{\cancel{2}\times 1}\times\frac{4\times 3\times\cancel{2}\times 1}{10\times 9\times\cancel{8}\times\cancel{7}} = \frac{12}{90} \approx 0.13.$$

Problem 4 (10 marks) Graph Theory

(a) Let G be the graph shown below. Use set-roster notation to write down a set $V(G)$ and a multiset of size-2 multisets $E(G)$ that together describe G, and also write down an adjacency matrix that describes G.



$V(G) = \{A, B, C, D, E\}$

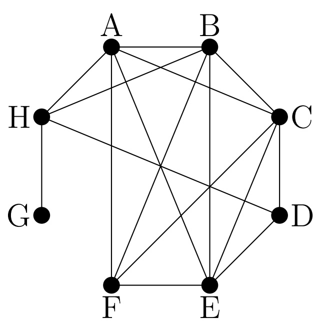
$E(G) = \{\{A,B\}, \{B,C\}, \{B,C\}, \{C,D\}, \{E,E\}\}$.

The following adjacency matrix describes G

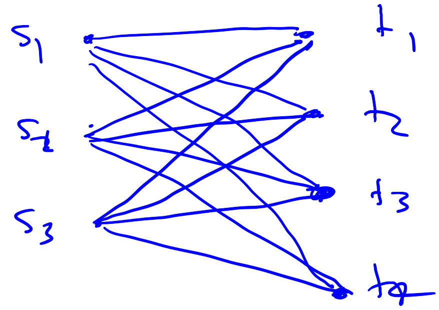
$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix} \end{array}$$

1 also accepted

(b) Let M be the graph shown below. Prove that M does not have a subgraph isomorphic to $K_{3,4}$.



Sps that M contains a subgraph S isomorphic to $K_{3,4}$. Then S is



Since G has degree 1 in M, it can't be in S. So the vertices of S are $\{A, B, C, D, E, F, H\}$. Since D has degree 3 in G, and D is adjacent to C, E, H we have that $D \in \{t_1, t_2, t_3, t_4\}$ and $\{C, E, H\} = \{s_1, s_2, s_3\}$. Hence $\{A, B, D, F\} = \{t_1, t_2, t_3, t_4\}$. But H is not adjacent to F. This is a contradiction.

(c) Let $n \in \mathbb{N}$. The hypercube of dimension n, denoted H_n, is the simple graph such that:

- $V(H_n)$ is the set of length-n bit strings;
- two length-n bit strings are adjacent if and only if the bit-strings differ in exactly one position.

(i) How many vertices does H_n have?

$|V(H_n)| = \#$ length-n bit strings $= 2^n$

(ii) What is the degree of each vertex in H_n?

degree of each vertex $= \#$ bits that could change $= n$

(iii) How many edges does H_n have?

$2|E(H_n)| =$ Total degree of $H_n = n2^n$

By the Handshaking Lemma By (i) and (ii)

So $|E(H_n)| = \frac{n2^n}{2} = n2^{n-1}$.

(iv) If there exists a Hamilton circuit in H_4, write one down; if no Hamilton circuit exists in H_4, explain how you know.

We construct a Hamilton circuit in H_4 using a reflected binary code.

0000
0001
0011
0010
0110
0111
0101
0100
1100
1101
1111
1110
1010
1011
1001
1000 $\to$ 0000.

Problem 5 (10 marks) (a) Describe the input and output of Dijkstra's algorithm.

INPUT: Connected simple graph G.
Vertices A, Z from G
Distance function $E(G) \to \mathbb{Q}^+$

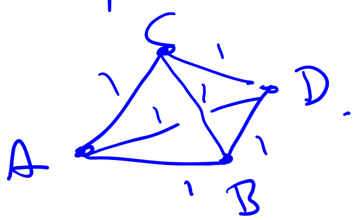
OUTPUT (a) Tree T containing A, Z as vertices and such that T is a subgraph of G and the unique path from A to Z in T has minimal length among all paths from A to Z in G. (b) Labelling $L: V(T) \to \mathbb{Q}^+$ such that $L(v) =$ min. distance $(A \to v)$ in G.

(b) Let S be the set of connected weighted simple graphs with 4 vertices. Prove or disprove the following statement:

$$\forall G \in S \;\; \exists!\, T \in S \;\; (T \text{ is a minimal spanning tree of } G).$$

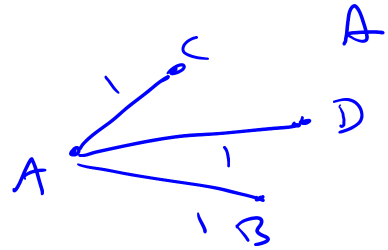
The following example shows the statement to be false.

Let G be



and

$T_1 =$



and T_2

. Then T_1 and T_2 are both minimal spanning trees of G.

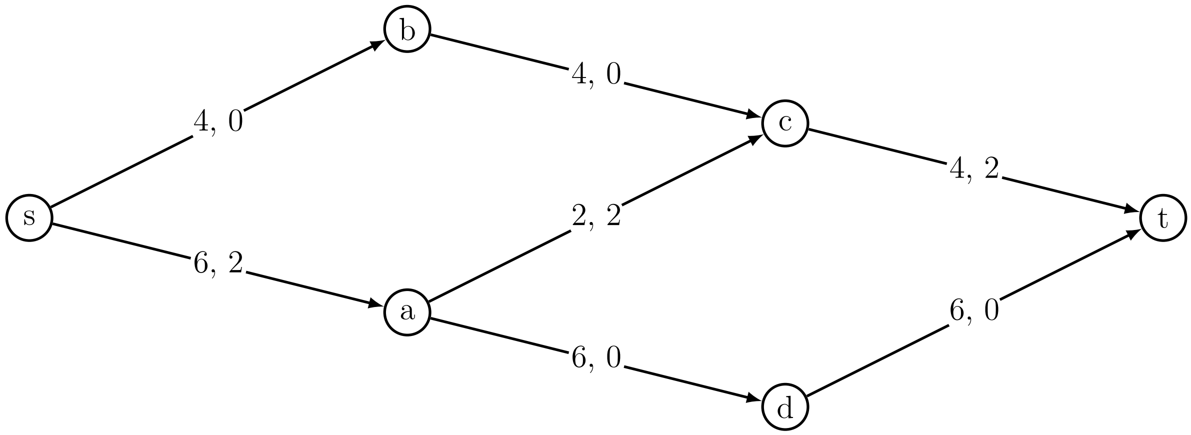
(c) Suppose that you are given a weighted digraph G that represents a transport network, and your colleague says "At any time, we can make at most 20 units flow through this network." Your friend then shows you an example of a flow of volume 20 across the network.

One way to check your colleague's statement is to apply one of the algorithms we learned in the course. Describe another method by which you may determine whether your colleagues statement is true or false, and justify why your method will allow you to be confident in your answer.

Find a cut with total weight 20.

By the min cut/max flow theorem, this confirms that the flow my colleague found is a max flow. If no cut of weight 20 exists, my colleague is wrong.

(d) Use the vertex labelling algorithm described in the course to find a maximum flow function for the transport network shown below (pseudocode for this algorithm is given at the end of the exam paper). The first incremental flow f_1 is shown in the first row of the table at the bottom of the page, and the cumulative flow F_1 is shown in the graph. Write down the subsequent incremental flows in the table (use only as many rows as you need).



incremental flow label	path of incremental flow	volume of incremental flow
f_1	$s\ a\ c\ t$	2
f_2	sbct	2
f_3	sadt	4
f_4	sbcadt	2

a virtual flow

Problem 6 (10 marks) (a) In no more than three sentences, explain how an internet is modelled by some type of graph (called a webgraph) for the purposes of the PageRank algorithm. An excellent answer will detail what type of graph is used, what the vertices represent, and what the edges represent?

A webgraph is a simple digraph in which vertices represent pages and a directed edge from vertex X to vertex Y represents a hyperlink on page X that goes to page Y.

(b) The PageRank algorithm may be understood to be following the movement of "The Random Surfer". In no more than five sentences, explain how The Random Surfer moves around the internet.

1) RS chooses a page on the web at random to start on.

2) At each step, RS moves to a page according to the following

2.1) with probability α, the RS will type in the URL of a randomly chosen page (possibly the current page) - this is called unforced teleporting.

2.2) with probability $(1-\alpha)$, the RS will proceed as follows - If there is at least one hyperlink on the current page, the RS selects one of these at random and clicks on it; o/w the RS will choose a new page at random from the web and type in the URL

(forced teleportation)

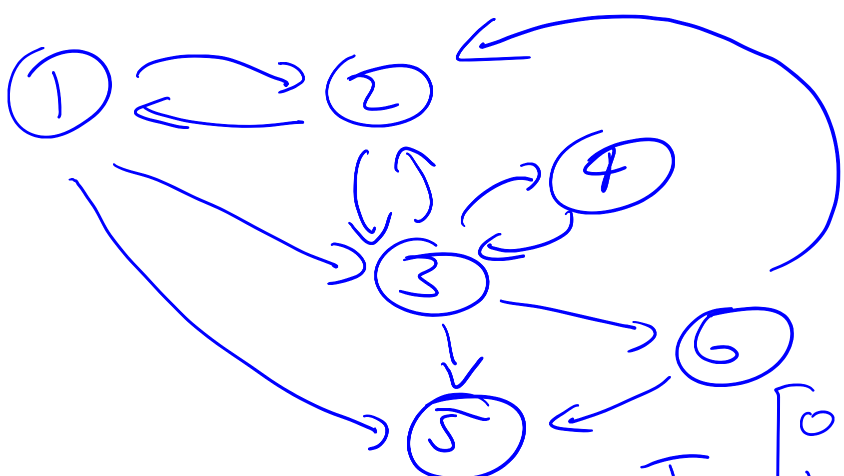
(c) Let $n \in \mathbb{N}$, let G be a webgraph with n vertices, let $\alpha = 0.15$ and let M denote the modified transition matrix (in the PageRank algorithm) determined by G and α. In the box below, write down an equation that completes the definition of the PageRank vector PR for G and α.

The PageRank vector PR is the unique $n{\times}1$ matrix such that its entries sum to 1 and the following equation is satisfied

$$\boxed{M'(\mathit{PR}) = (\mathit{PR}).}$$

(d) Let G be the webgraph with the adjacency matrix A shown below. Draw a picture of the graph G represented by A, and then write down the basic transition matrix T associated to A as part of the PageRank algorithm.

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$



$$T = \begin{bmatrix} 0 & 1/3 & 1/3 & 0 & 1/3 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1/4 & 0 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1/5 & 1/5 & 1/5 & 1/5 & 0 & 1/5 \\ 0 & 1/2 & 0 & 0 & 1/2 & 0 \end{bmatrix}$$

THIS IS THE END OF THE EXAM. PSEUDOCODE FOR THE VERTEX LABELLING ALGORITHM IS OVER THE PAGE.

Vertex labelling algorithm for finding a maximum flow function for a transport network

Input: Transport network D with capacity function C.

Output: A maximum flow function $F_{\max}$ for the network.

Method: Initialise F to the zero flow F_0. Initialize i to 1.

For $i = 1, 2, \ldots$ carry out stage i below to attempt to build an incremental flow f_i.

If stage i succeeds, define $F_i = F_{i-1} + f_i$ and proceed to stage $i+1$.

If stage i fails, define $F_{\max} = F_{i-1}$ and stop.

Stage i:

1. If $i > 1$, mark up the amended edge flows for F_{i-1}.
2. Mark up the levels for F_{i-1}, as explained below.
3. If t is assigned a level, stage i will succeed, so continue.

   If not, then stage i fails, so return above to define $F_{\max}$ and terminate.
4. Mark up labels for F_{i-1} as follows until t is labelled:
   (a) Label each level 1 vertex v with sk_{v}, where $k_{\mathrm{v}} = S((s,v))$. (see below for definition of S)
   (b) If t has level 2 or more now work through the level 2 vertices in alphabetical order, labelling each vertex v with uk_u, where
      - u is the alphabetically earliest level 1 vertex with $(u,v) \in E(D)$ and $S((u,v)) > 0$,
      - k_{v} is the minimum of $S((u,v))$ and the value part of u's label.

   (c) If t has level 3 or more now work through the level 3 vertices in a similar manner and so on.
5. Let p_i be the path $\mathrm{u}_0\mathrm{u}_1 \ldots \mathrm{u}_n$ where $\mathrm{u}_n = \mathrm{t}$ and for $0 < j \leq n$ u_j has label $\mathrm{u}_{j-1}k_j$.

   Define f_i to be the incremental flow on p_i with flow value k_n.

End of Method

Levels and labels: At each stage of the vertex labelling algorithm levels and labels are associated afresh with the vertices of the network.

The **level** of a vertex is determined iteratively as follows:

- The source vertex s always has level 0.
- If $e = (s, x)$ and $S(e) > 0$ then x has level 1.
- If x has level n and $S((x, y)) > 0$ then y has level $n + 1$ provided it has not already been assigned a lower level.

The **label** on a vertex v of level $n \geq 1$ has the form uk, where u is a vertex of level $n - 1$ and $(u, v) \in E(D)$ is an edge on the path for a potential incremental flow through v with flow value k.

The algorithm assigns labels in ascending order of levels, and in alphabetical order within levels.

The spare capacity function S
For vertices u,v of D, where D has capacity and flow functions C, F:

$$S((\mathrm{u,v})) = \begin{cases} C((\mathrm{u,v})) - F((\mathrm{u,v})) & \text{if } (\mathrm{u,v}) \in E(D) \\ F((\mathrm{v,u})) & \text{if } (\mathrm{v,u}) \in E(D) \\ 0 & \text{otherwise.} \end{cases}$$

When (v,u) $\in E(D)$, $S((\mathrm{u, v}))$ is called a **virtual capacity**.