Duration: **50 minutes**
Aids Allowed: **NONE** (in particular, no calculator)

Student Number: └─┴─┴─┴─┴─┴─┴─┴─┴─┴─┘

Last (Family) Name(s): ______________________________

First (Given) Name(s): ______________________________

Do ***not*** *turn this page until you have received the signal to start.*
In the meantime, please read the instructions below *carefully*.

This term test consists of 3 questions on 10 pages (including this one), printed on both sides of the paper. *When you receive the signal to start, please make sure that your copy of the test is complete, fill in the identification section above, write your student number where indicated at the bottom of every odd-numbered page (except page 1), and write your name on the back of the last page.*

Answer each question directly on the test paper, in the space provided, and use the reverse side of the pages for rough work. If you need more space for one of your solutions, use the reverse side of a page and *indicate clearly the part of your work that should be marked.*

In your answers, you may use without proof any result or theorem covered in lectures, tutorials, homework, tests, or the textbook, as long as you give a clear statement of the result(s)/theorem(s) you are using. You must justify all other facts required for your solutions.

Write up your solutions carefully! In particular, use notation and terminology correctly and explain what you are trying to do — part marks *will* be given for showing that you know the general structure of an answer, even if your solution is incomplete.

If you are unable to answer a question (or part), you will get 20% of the marks for that question (or part) if you write “I don’t know” and nothing else — you will *not* get those marks if your answer is completely blank, or if it contains contradictory statements (such as “I don’t know” followed or preceded by parts of a solution that have not been crossed off).

Marking Guide

# 1: ______/10

# 2: ______/16

3: ______/ 7

TOTAL: ______/33

Good Luck!

Use this page for rough work — clearly indicate any section(s) to be marked.

Question 1. [10 MARKS]

Consider the following task: given a list that contains stock prices over the last several days, determine if there are any three consecutive days during which the stock price increased. Write a divide-and-conquer algorithm that carries out this task; more specifically, give an algorithm $A(L, b, e)$ that returns an index i such that $b + 1 \leqslant i \leqslant e - 1$ and $L[i - 1] < L[i] < L[i + 1]$, if such an index exists—your algorithm should return the value -1 if there is no such index. Include descriptive comments, or a concise English description of the main ideas, and analyze the running time of your algorithm. (For your reference, the Master Theorem states that a recurrence of the form $T(n) = aT(n/b) + \Theta(n^d)$ has solution $\Theta(n^d)$ if $a < b^d$, $\Theta(n^d \log n)$ if $a = b^d$, and $\Theta(n^{\log_b a})$ if $a > b^d$.)
Note: For full marks, your answer must make use of the divide-and-conquer method (even though there is a simple iterative solution), and your algorithm must run in no more than linear time.

Use this page for rough work — clearly indicate any section(s) to be marked.

CONT'D...

Question 2. [16 MARKS]

A *vertex cover* in an undirected graph $G = (V, E)$ is any subset of vertices $C \subseteq V$ such that every edge in G has at least one endpoint in C (*i.e.*, $\forall\ (u, v) \in E$, $u \in C$ or $v \in C$, or both). In the *minimum vertex cover* problem, the input is a graph with a positive integer weight $w(v)$ for each vertex $v \in V$, and we want to find a vertex cover C with the smallest possible total weight $w(C) = \sum_{v \in C} w(v)$. In this question, we are interested in constructing minimum weight vertex covers for trees, using dynamic programming.

Part (a) [3 MARKS]

Given a tree $T = (V, E)$, state precisely the information contained in your dynamic programming table(s)—in particular, specify the exact sub-problem whose solution is stored in each entry of your table(s).

Part (b) [6 MARKS]

Give a recurrence that could be used to compute the entries in your table. Justify that your recurrence is correct.

Use this page for rough work — clearly indicate any section(s) to be marked.

Question 2. (CONTINUED)

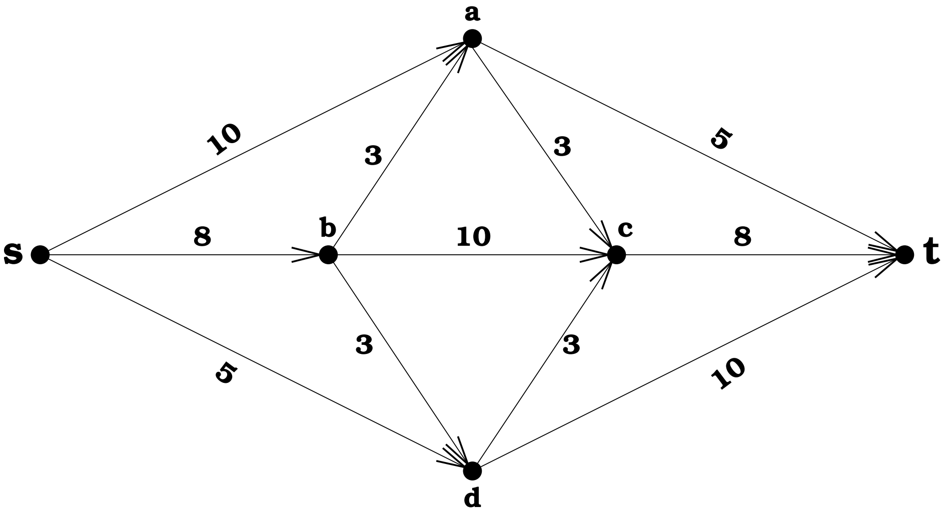
Part (c) [7 MARKS]

Give an algorithm that will determine the vertices in a minimum vertex cover, given the entries in your table. Briefly justify that your algorithm is correct, and analyze its runtime.

Use this page for rough work — clearly indicate any section(s) to be marked.

Question 3. [7 MARKS]

Use the Ford-Fulkerson algorithm to compute a maximum flow in the network below (where each edge has the indicated capacity). State precisely each augmenting path that you use—you **must** use the path given below the picture as your first augmenting path. Justify that your flow is maximum—for full marks, your justification should *not* be based on the non-existence of a larger flow or of an augmenting path.



Augmenting path $P = s \xrightarrow{0/8} b \xrightarrow{0/10} c \xrightarrow{0/8} t$ Residual capacity $\Delta_f(P) =$ ____ (*fill this in*)

On this page, please write nothing except your name.

Last (Family) Name(s): ____

First (Given) Name(s): ____