M269 Exams

Prsntn 2013J Exam

Contents

1 M269 Prsntn 2013J Exam Qs

1.1 M269 2013J Exam Qs

- M269 Algorithms, Data Structures and Computability
- Presentation 2013J Exam
- Date Wednesday, 4 June 2015 Time 14:30–17:30
- There are **TWO** parts to this examination. You should attempt all questions in **both** parts
- **Part 1** carries **60 marks** — **100 minutes**
- **Part 2** carries **40 marks** — **70 minutes**
- **Note** see the original exam paper for exact wording and formatting — these slides and notes may change some wording and formatting

Go to Exam Solns

1.2 M269 2013J Exam Q Part1

- Answer every question in this part.
- Answers to questions in this part should be written on this paper in the spaces provided, or in the case of multiple-choice questions you should tick the appropriate box(es).
- If you tick more boxes than indicated for a multiple choice question, you will receive **no** marks for your answer to that question.

Go to Exam Soln Part1

1.3 M269 2013J Exam Q 1

- Which **two** of the following statements are **true**?

A. A decision problem is any problem stated in a formal language.

B. A computational problem is a problem that is expressed sufficiently precisely that it is possible to build an algorithm that will solve all instances of that problem.

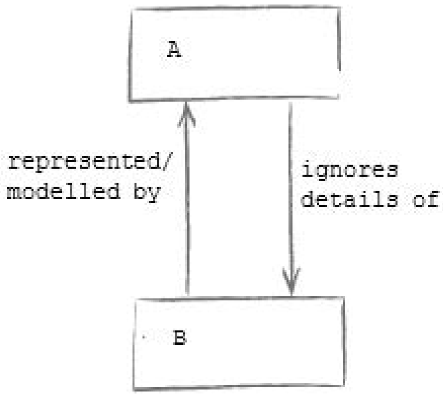
C. An algorithm consists of a precisely stated, step-by-step list of instructions.

D. Computational thinking is the skill to formulate a problem as a computational problem, and then construct a good computational solution, in the form of an algorithm, to solve this problem, or explain why there is no such solution. **(2 marks)**

Go to Exam Soln 1

1.4 M269 2013J Exam Q 2

- **Question 2** The general idea of abstraction as modelling can be shown with the following diagram. **(2 marks)**



- Complete the diagram above by adding an appropriate label (one of the numbers 1 to 4) in the space indicated by **A** and one in the space indicated by **B**. The possible answers are shown as 1 to 4 below.

  1. A car crash test dummy in the real world
  2. An action man doll in the real world
  3. A real car in the real world (after crashing)
  4. A real driver in the real world

- *The exam question had some pictures next to the texts*

Go to Exam Soln 2

1.5 M269 2013J Exam Q 3

- **Question 3** A binary search is being carried out on the list shown below for item 67: **(4 marks)**

  [12,16,17,24,41,49,51,62,67,69,75,80,89,97,101]

- For each pass of the algorithm, draw a box around the items in the partition to be searched during that pass, continuing for as many passes as you think are needed.
- We have done the first pass for you showing that the search starts with the whole list. Draw your boxes below for each pass needed; you may not need to use all the lines below. *(The question had 8 rows)*

(Pass 1) [$\boxed{12,16,17,24,41,49,51,62,67,69,75,80,89,97,101}$]
(Pass 2) [12,16,17,24,41,49,51,62,67,69,75,80,89,97,101]
(Pass 3) [12,16,17,24,41,49,51,62,67,69,75,80,89,97,101]

Go to Exam Soln 3

1.6 M269 2013J Exam Q 4

- **Question 4** Consider the guard in the following Python while loop header: **(4 marks)**

```
while (a < 6 and b > 8) or not(a >= 6 or b <= 8):
```

(a) Make the following substitutions:

P represents `a < 6`

Q represents `b > 8`

Then complete the following truth table:

P	Q	$\neg P$	$\neg Q$	$P \wedge Q$	$\neg P \vee \neg Q$	$\neg(\neg P \vee \neg Q)$	$(P \wedge Q) \vee \neg(\neg P \vee \neg Q)$
F	F						
F	T						
T	F						
T	T						

(b) Use the results from your truth table to choose which one of the following expressions could be used as the simplest equivalent to the above guard.

A. `(a < 6 and b > 8)`

B. `not(a < 6 and b > 8)`

C. `(a >= 6 or b <= 8)`

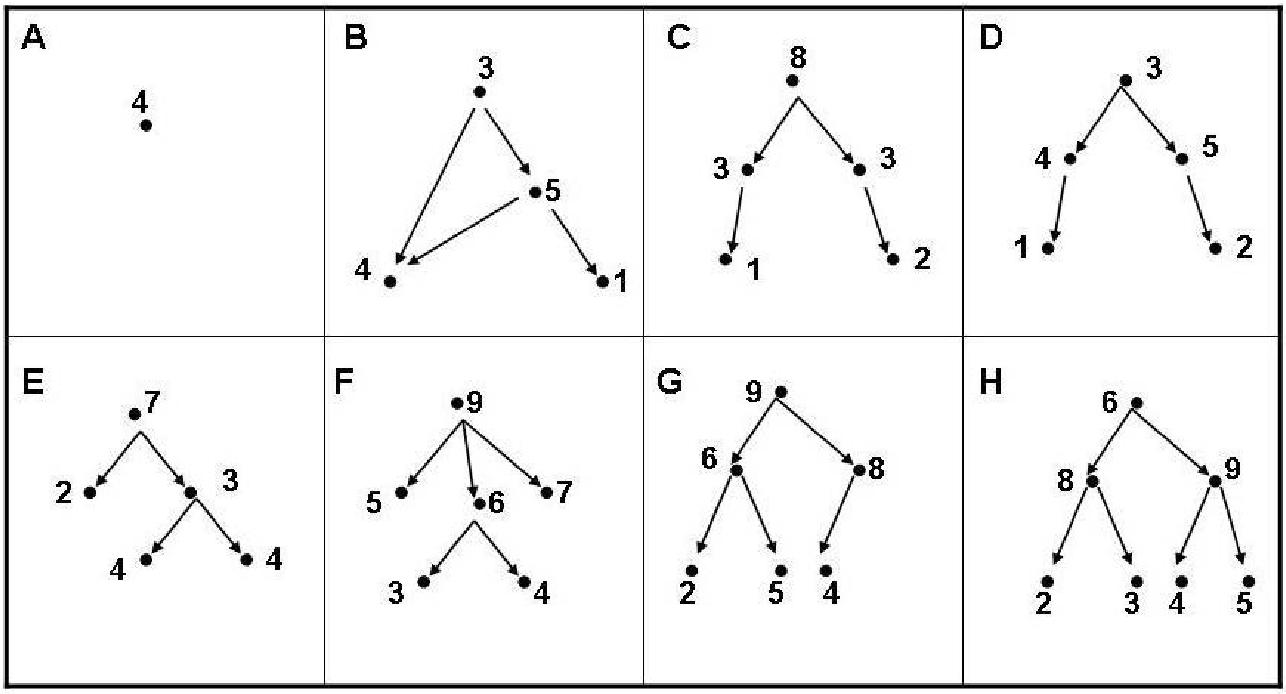
D. `(a >= 6 and b <= 8)`

E. `(a < 6 and b <= 8)`

Go to Exam Soln 4

1.7 M269 2013J Exam Q 5

- **Question 5** Consider the following diagrams A–H. Nodes are represented by black dots and edges by arrows. The numbers represent a node's key. **(4 marks)**



- Answer the following questions. Write your answer on the line that follows each question. In each case there is at least one diagram in the answer but there may be more than one. Explanations are not required.

(a) Which of A, B, C and D **do not** show trees ?

(b) Which of E, F, G and H are binary trees ?

(c) Which of C, D, G and H are complete binary trees ?

(d) Which of C, D, G and H are binary heaps ?

Go to Exam Soln 5

1.8 M269 2013J Exam Q 6

- Consider the following function, which takes a list as an argument. You may assume that the list contains a number of integer values and is not empty.

```
def average(aList):
    n = len(aList)
    total = 0
    for item in aList:
        total = total + item
    mean = total / n
return mean
```

- From the five options below, select the **one** that represents the correct combination of T(n) and Big-O complexity for this function. You may assume that a step (i.e. the basic unit of computation) is the assignment statement.

A. $T(n) = 3 + n^2$ and $O(n^2)$

B. $T(n) = n + 2$ and $O(n^2)$

C. $T(n) = 2n + 2$ and $O(n)$

D. $T(n) = 3n + n^2$ and $O(n^2)$

E. T(n) = n + 3 and O(n)

Go to Exam Soln 6

1.9 M269 2013J Exam Q 7

- In the KMP algorithm, for each character in the target string *T* we identify the longest substring of *T* ending with that character which matches a prefix of the target string.
- These lengths are stored in what is known as a **prefix table** (which in Unit 4 we represented as a list).
- Consider the target string *T*

A	B	A	C	A	B	A	C	A	C

- Below is an incomplete prefix table for the target string given above. Complete the prefix table by writing the missing numbers in the appropriate boxes.

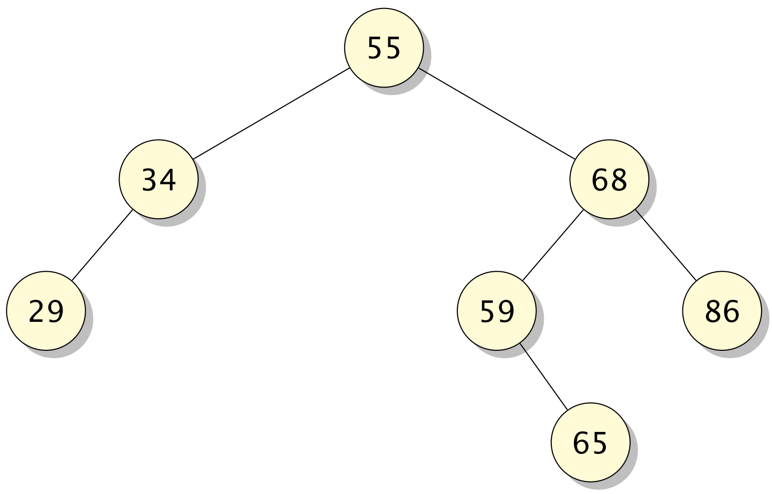
0		1	0		2		4		0

Go to Exam Soln 7

1.10 M269 2013J Exam Q 8

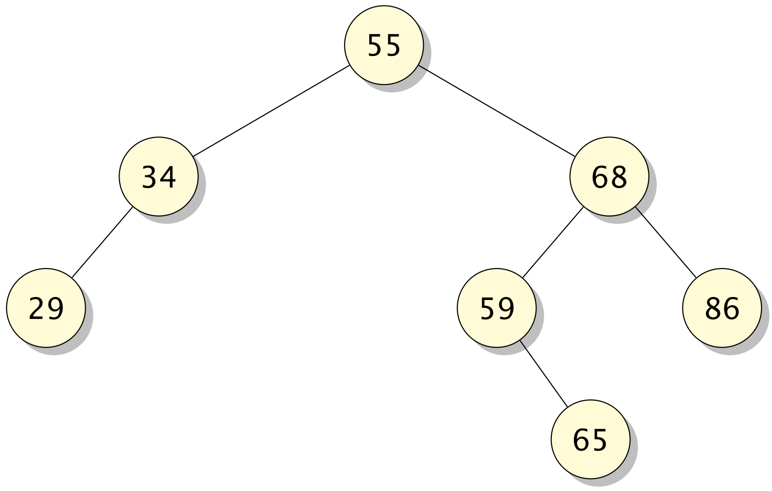
- **Question 8** **(4 marks)**

(a) Consider the following Binary Search Tree.



- Modify (draw on) the above Binary Search Tree to insert a node with a key of 57.

(b) Once again, consider the same Binary Search Tree.



- Calculate the balance factors of each node in the tree above and modify the diagram to show these balance factors.

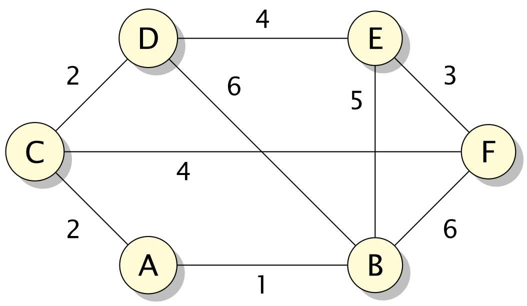
Go to Exam Soln 8

1.11 M269 2013J Exam Q 9

- Recall that the structured English for Dijkstra's algorithm is:

```
create priority~queue
set dist to 0 for v and dist to infinity
  for all other vertices
add all vertices to priority~queue
ITERATE while priority~queue is not empty
  remove u from the front of the queue
  ITERATE over w in the neighbours of u
    set new~distance to
      dist u + length of edge from u to w
    IF new~distance is less than dist w
      set dist w to new~distance
      change priority(w, new~distance)
```

- Now consider the following weighted graph:



- **Starting from vertex B**, the following table represents the distances from each vertex to A after the second line of structured English is executed for the graph given above (using the convention that a blank cell represents infinity):

Vertex	A	B	C	D	E	F
Distance		0				

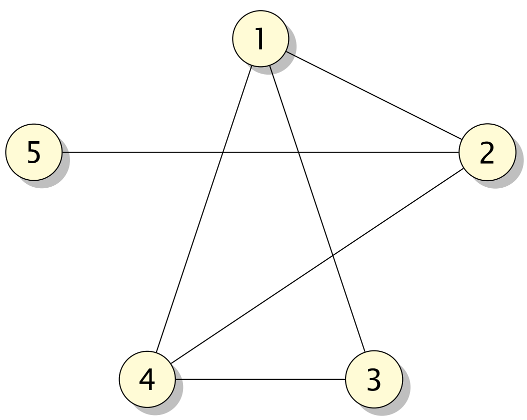
- *Note that neither the table above nor the subsequent tables represent the priority queue.*
- Now, complete the appropriate boxes in the next table to show the distances after the first and second iterations of the `while` loop of the algorithm.

Vertex	A	B	C	D	E	F	
Distance		0					First iteration
Distance		0					Second iteration

Go to Exam Soln 9

1.12 M269 2013J Exam Q 10

- **Question 10** Consider the following graph: **(4 marks)**



- Complete the table below to show the order in which the vertices of the above graph could be visited in a Depth First Search (DFS) starting at vertex 3 and always choosing first the leftmost not yet visited vertex (as seen from the current vertex):

Vertex	3				

Go to Exam Soln 10

1.13 M269 2013J Exam Q 11

- **Question 11** **(4 marks)**

(a) What does it mean to say that a well-formed formula (WFF) is *satisfiable* ? Use the space below for your answer

(b) Is the following WFF satisfiable ?

$(P \rightarrow (Q \rightarrow P)) \vee \neg R$

- Explain how you arrived at your answer

Go to Exam Soln 11

1.14 M269 2013J Exam Q 12

- A particular interpretation of predicate logic allows facts to be expressed about films that people have seen, and of which they own copies.
- Some of the assignments in the interpretation are given below (where the symbol $\mathcal{I}$ is used to show assignment).
- The interpretation assigns *Jane*, *John* and *Saira* to the constants *jane*, *john* and *saira*.

$$\begin{array}{rcl} \mathcal{I}(\text{jane}) & = & \text{Jane} \\ \mathcal{I}(\text{john}) & = & \text{John} \\ \mathcal{I}(\text{saira}) & = & \text{Saira} \end{array}$$

- The predicates *owns* and *has_seen* are assigned to binary relations. The comprehensions of the relations are:
  - $\mathcal{I}$(owns) = {(A,B): the person A owns a copy of film B}
  - $\mathcal{I}$(has_seen) = {(A,B): the person A has seen film B}
- The enumerations of the relations are:

$\mathcal{I}$(owns) = {(Jane, Django), (Jane, Casablanca), (John, Jaws), (John, The Omen), (John, El Topo), (Saira, El Topo), (Saira, Casablanca)}

$\mathcal{I}$(has_seen) = {(Jane, Django), (Jane, Candide), (Jane, Casablanca), (John, The Omen), (John, El Topo), (Saira, Django), (Saira, The Omen)}

- Parts (a) and (b) of this question are on the next page.
- In both parts, you are given a sentence of predicate logic and asked to provide an English translation of the sentence in the box immediately following it.
- You also need to state whether the sentence is TRUE or FALSE in the interpretation that is provided on this page, and give an explanation of your answer.
- In your explanation you need to consider any relevant values for the variable X, *and show, using the interpretation above, whether it makes the quantified expression TRUE.*

(a) $\forall \text{X}.(owns(saira, \text{X}) \rightarrow has\_seen(saira, \text{X}))$

can be translated in English as:

- This sentence is TRUE/FALSE because:

(b) $\exists \text{X}.(has\_seen(jane, \text{X}) \wedge owns(jane, \text{X}))$

can be translated in English as:

- This sentence is TRUE/FALSE because:

Go to Exam Soln 12

1.15 M269 2013J Exam Q 13

- **Question 13** A database contains the following tables, *oilfield* and *operator* **(6 marks)**

oilfield	
name	*production*
Warga	3
Lolli	5
Tolstoi	0.5
Dakhun	2
Sugar	3

operator	
company	*field*
Amarco	Warga
Bratape	Lolli
Rosbif	Tolstoi
Taqar	Dakhun
Bratape	Sugar

- For each of the following SQL queries, give the table returned by the query

(a)
```sql
SELECT *
FROM operator;
```
(b)
```sql
SELECT name, production
FROM oilfield
WHERE production > 2;
```
(c)
```sql
SELECT name, production, company
FROM oilfield CROSS JOIN operator
WHERE name = field;
```

Go to Exam Soln 13

1.16 M269 2013J Exam Q 14

- **Question 14** Consider the following axiom schema and rules: **(4 marks)**

Axiom schema	$\{\mathbf{A}\} \vdash \mathbf{A}$	
Rules	$\dfrac{\Gamma \vdash \mathbf{A} \wedge \mathbf{B}}{\Gamma \vdash \mathbf{A}}$	($\wedge$-elimination left)
	$\dfrac{\Gamma \vdash \mathbf{A} \wedge \mathbf{B}}{\Gamma \vdash \mathbf{B}}$	($\wedge$-elimination right)
	$\dfrac{\Gamma \vdash \mathbf{A} \quad \Gamma \vdash \mathbf{B}}{\Gamma \vdash \mathbf{A} \wedge \mathbf{B}}$	($\wedge$-introduction)
	$\dfrac{\Gamma \cup \{\mathbf{A}\} \vdash \mathbf{B}}{\Gamma \vdash \mathbf{A} \rightarrow \mathbf{B}}$	($\rightarrow$-introduction)
	$\dfrac{\Gamma \vdash \mathbf{A} \quad \Gamma \vdash \mathbf{A} \rightarrow \mathbf{B}}{\Gamma \vdash \mathbf{B}}$	($\rightarrow$-elimination)

- Complete the following proof by filling in the two boxes. You can use any of the above as appropriate.

1.	$\{P \wedge (Q \wedge R)\} \vdash P \wedge (Q \wedge R)$	[Axiom schema]
2.	$\boxed{?? \qquad\qquad ??}$	[1,$\wedge$-elimination left]
3.	$\emptyset \vdash (P \wedge (Q \wedge R)) \rightarrow P$	$\boxed{?? \qquad\qquad ??}$

Go to Exam Soln 14

1.17 M269 2013J Exam Q 15

- **Question 15** Which *two* of the following statements are true? **(2 marks)**

A. A Turing Machine is a mathematical model of computational problems.

B. If the lower bound for a computational problem is $O(n^2)$, then there is an algorithm that solves the problem and which has complexity $O(n^2)$.

C. Searching a sorted list is not in the class NP.

D. The decision Travelling Salesperson Problem is NP-complete.

E. There is no known tractable quantum algorithm for solving a known NP-complete problem.

Go to Exam Soln 15

1.18 M269 2013J Exam Q Part2

- Answer every question in this part.
- The marks for each question are given at the end of the question.
- Answers to this part should be written in the separate answer book

Go to Exam Soln Part2

1.19 M269 2013J Exam Q 16

- Multipart question

- Specification of program, data structures, pre and post conditions
- Write a small program
- Give the complexity of the small program
- Give insight into a sorting algorithm
- Give insight into insertion into a binary search tree
- See notes version for text

Go to Exam Soln 16

1.19.1 M269 2013J Exam Q 16 Text

The Widget & Widget Widget Corporation (W&WWC) keeps records of every client that has purchased widgets from them, along with details of the value of every purchase. These records are stored on a computer in two sequences; the first of these, CLIENTS, is an (unsorted) sequence of the clients' names; the second, SPENDS, contains a sequence of sequences, with each item representing the sequence of values of each of the purchases that a client has made. The index of a client in CLIENTS is the index of that client's sequence of purchases in SPENDS.

(a) W&WC requires a small computer program which will provide them with two facilities:

- one to return the average spend of a specified client, if that client is present in the data, and to return a suitable value if the client is not present;
- the other to return a sequence containing for each client in CLIENTS their average spend.

(i) Express both as a computational problem by completing the templates below (in your answer book). Make whatever decisions you think appropriate about the exact form of the input and output.

Name: `SpecifiedClientAverageSpend`

Inputs:

Outputs:

Name: `EachClientAverageSpend`

Inputs:

Inputs:

(ii) In addition, suggest one possible postcondition for the `SpecifiedClientAverageSpend` problem.

(iii) Provide the structured English for the `EachClientAverageSpend` algorithm. (If you wish you can write a Python function instead, but not both.)

(b) What will be the complexity, expressed in T(n, q) and Big-O format, of your `EachClientAverage` solution, assuming n clients with an average of q transactions each, and that the assignment statement is the unit of computation? Explain your reasoning.

(c) One of the drawbacks of the current way in which the data is stored is that the sequence of clients is not sorted. One of the best ways of sorting a sequence is the *Quicksort* algorithm. Express your understanding of this algorithm for in-place sorting in the form of an initial insight.

(d) Having developed the current program as far as they can using sequences, managers have now made the decision to store clients and transactions in a *Binary Search Tree* (BST).

Express the initial insight for an algorithm for inserting a new node N, with key k, into a BST.

1.20 M269 2013J Exam Q 17

- Write short report on a computational topic
- Suitable title for the topic and audience
- Paragraph setting the scene — the context of the topic
- Paragraph describing the topic
- Paragraph on the role the topic plays in some area
- Conclusions justifying the importance of the topic
- See notes version for text.

Go to Exam Soln 17

1.20.1 M269 2013J Exam Q 17 Text

Imagine that you are a potential speaker for your local University of the Third Age (U3A). Along with other potential speakers you've been asked to write a short report on a particular topic. The organising group will then look at these reports and choose which potential speaker to ask for a full evening presentation. Your topic is *The Turing Machine*. Write a short report. Your report must have the following structure:

1. A suitable title.
2. A paragraph *setting the scene* and explaining the historical importance of the deterministic Turing Machine *[about three sentences]*.
3. One paragraph in which you describe in layperson's terms what a deterministic Turing Machine is *[about three sentences]*.
4. One paragraph in which you describe the role that Turing Machines play in Turing's proof that there are computational problems that are not computable *[about three sentences]*.
5. A conclusion in which you give a reasoned conclusion about the importance of Turing Machines *[one sentence]*.

Note that a significant number of marks will be awarded for coherence and clarity, so avoid abrupt changes of topic and make sure your sentences fit together to tell an *overall* story. Allow up to four additional sentences to ensure this.

2 M269 Prsntn 2013J Exam Solns

2.1 M269 2013J Exam Solns

- The solutions given below are not official solutions
- For some questions, alternatives are given — a student would only have to provide one

Go to Exam Qs

2.2 M269 2013J Exam Soln Part1

- Part 1 solutions

Go to Exam Q Part1

2.3 M269 2013J Exam Soln 1

- Options C and D are true.
- Option A is wrong because decision problems have to have a yes-no answer.
- Option B is wrong because there are computational problems that we can state and build algorithms for, but cannot always be solved.

Go to Exam Q 1

2.4 M269 2013J Exam Soln 2

- A real driver is modelled by a car crash test dummy, so A = 1 and B = 4

Go to Exam Q 2

2.5 M269 2013J Exam Soln 3

- The complete binary search:

  (Pass 1) [$\boxed{12,16,17,24,41,49,51,62,67,69,75,80,89,97,101}$]
  (Pass 2) [12,16,17,24,41,49,51,62,$\boxed{67,69,75,80,89,97,101}$]
  (Pass 3) [12,16,17,24,41,49,51,62,$\boxed{67,69,75}$,80,89,97,101]
  (Pass 4) [12,16,17,24,41,49,51,62,$\boxed{67}$,69,75,80,89,97,101]

Go to Exam Q 3

2.6 M269 2013J Exam Soln 4

(a) The completed truth table:

P	Q	¬P	¬Q	P ∧ Q	¬P ∨ ¬Q	¬(¬P ∨ ¬Q)	(P ∧ Q) ∨ ¬(¬P ∨ ¬Q)
F	F	T	T	F	T	F	F
F	T	T	F	F	T	F	F
T	F	F	T	F	T	F	F
T	T	F	F	T	F	T	T

(b) A is the simplest equivalent of the guard given (and the only equivalent).

Go to Exam Q 4

2.7 M269 2013J Exam Soln 5

(a) B is not a tree; it has more than one route from node 3 to node 4.

(b) E, G, and H are binary trees; (no more than 2 children per node).

(c) G, and H are complete binary trees.

(d) Only G is a heap; (complete binary tree, and parent nodes > children).

Go to Exam Q 5

2.8 M269 2013J Exam Soln 6

- Option E is correct.
- The function does three assignments once per call, and one assignment for each of the n items in the argument, hence $T(n) = n + 3$.

Go to Exam Q 6

2.9 M269 2013J Exam Soln 7

- The complete prefix table, with new entries in blue:

0	0	1	0	1	2	3	4	5	0

- Here is the target, prefix and shift:

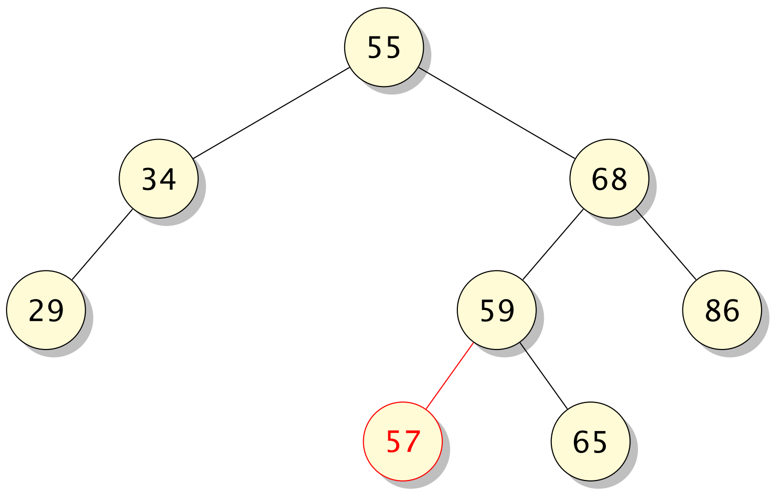
	A	B	A	C	A	B	A	C	A	C	Target string, t
	0	1	2	3	4	5	6	7	8	9	Position (Index), p
0	1	2	3	4	5	6	7	8	9	10	Match, q
	0	0	1	0	1	2	3	4	5	0	prefixTable(t, p)
1	1	2	2	4	4	4	4	4	4	10	shift(t, q)

- The shift function takes the *target string*, t, and the number of characters matched, q.
- shift(t, 0) = 1
- shift(t, q) = q − prefixTable(t, q − 1)

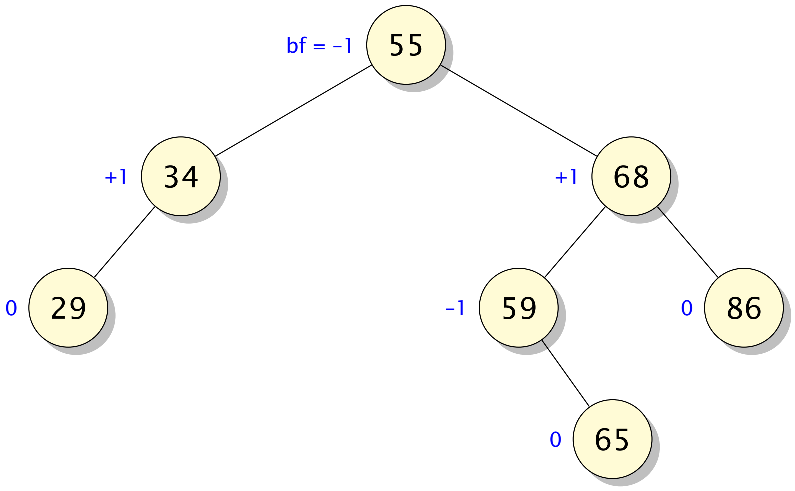
Go to Exam Q 7

2.10 M269 2013J Exam Soln 8

(a) Answer, with inserted node shown in red



(b) Answer, with balance factors shown in blue



Go to Exam Q 8

2.11 M269 2013J Exam Soln 9

- The completed table

Vertex	A	B	C	D	E	F	
Distance	1	0		6	5	6	First iteration
Distance	1	0	3	6	5	6	Second iteration

Go to Exam Q 9

2.12 M269 2013J Exam Soln 10

- Depth First Search (DFS) starting at vertex 3 and always choosing first the leftmost not yet visited vertex (as seen from the current vertex):

Vertex	3	4	1	2	5

- Notice the ambiguity about the term *leftmost* — an alternative view could have been:

Vertex	3	1	4	2	5

Go to Exam Q 10

2.13 M269 2013J Exam Soln 11

(a) A WFF is *satisfiable* if it is possible to find an interpretation that makes the formula true.

(b) Truth table for the WFF

P	Q	R	Q → P	P → (Q → P)	¬R	(P → (Q → P)) ∨ ¬R
F	F	F	T	T	T	T
F	F	T	T	T	F	T
F	T	F	F	T	T	T
F	T	T	F	T	F	T
T	F	F	T	T	T	T
T	F	T	T	T	F	T
T	T	F	T	T	T	T
T	T	T	T	T	F	T

- The truth table shows that the WFF (P → (Q → P)) ∨ ¬R is always true, so it satisfiable under any interpretation. But we don't need the whole truth table to prove this; the WFF is true for any interpretation in which R is false (for example).

Go to Exam Q 11

2.14 M269 2013J Exam Soln 12

(a) *For all films, if Saira owns a copy of the film, then Saira has seen the film.*

Or more idiomatically, *Saira has seen all of the films that she owns.*

- The sentence is *FALSE*, because the enumerations of the relations show that she owns a copy of *Casablanca*, but this is not one of the films that she has seen. She also owns a copy of *El Topo*, which she has not seen either, but we only need one counter-example to show that the sentence is false.

(b) *There exists a film, such that Jane has seen it and Jane owns it.*

Or more idiomatically, *Jane has seen at least one of the films that she owns*

- This sentence is *TRUE*. The enumerations show that she owns *Casablanca* and that she has seen it. *Django* also provides a sufficient example to show that the sentence is true.

Go to Exam Q 12

2.15 M269 2013J Exam Soln 13

(a) This is simply the whole *operator* table.

company	*field*
Amarco	Warga
Bratape	Lolli
Rosbif	Tolstoi
Taqar	Dakhun
Bratape	Sugar

(b) Retaining only the rows with *production* > 2

name	*production*
Warga	3
Lolli	5
Sugar	3

(c) Joining the tables

name	*production*	*company*
Warga	3	Amarco
Lolli	5	Bratape
Tolstoi	0.5	Rosbif
Dakhun	2	Taqar
Sugar	3	Bratape

Go to Exam Q 13

2.16 M269 2013J Exam Soln 14

1.	$\{P \land (Q \land R)\} \vdash P \land (Q \land R)$	[Axiom schema]
2.	$\boxed{\{P \land (Q \land R)\} \vdash P}$	[1,$\land$-elimination left]
3.	$\emptyset \vdash (P \land (Q \land R)) \to P$	$\boxed{[2,\to\text{-introduction}]}$

Go to Exam Q 14

2.17 M269 2013J Exam Soln 15

- Only D and E are true.
- A Universal Turing Machine can compute any *computable* sequence but there are well defined problems that are not computable. (So not A)
- A lower bound may be lower than any actual algorithm. (So not B)
- Every problem in P is in NP — we just do not know if P == NP (So not C)

Go to Exam Q 15

2.18 M269 2013J Exam Soln Part2

- Part 2 Solutions

Go to Exam Q Part2

2.19 M269 2013J Exam Soln 16

- See notes version for text

Go to Exam Q 16

M269 Exam 2013J Q 16 Sample Solution

(a)

(i)

Name: `SpecifiedClientAverageSpend`

Inputs: Client name as String

Outputs: If client is found: Average spend for client as Real else: None

Name: `EachClientAverageSpend`

Inputs: None

Inputs: List of average spends for each client as list of real

(ii) Returned average is a sensible size?

(iii)

```python
def EachClientAverageSpend():
    averages = list()
    for purchases in SPENDS:
        if len(purchases) > 0:
            average = sum(purchases)/len(purchases)
        else:
            average = None
        averages.append(average)
    return averages
```

(b) Assuming n clients with q transactions, and assuming (a) that the Python function `sum` does not create any extra assignments, and (b) that the Python append method counts as one assignment, we have one assignment to create the empty list of averages, one to create the average for each customer, and one more to append the average to the list, so $T(n, q) = 1 + 2n$.

On the other hand we might assume that `sum(purchases)` counted as q assignments, and in this case we would have $T(n, q) = 1 + 2qn$. In either case n remains the dominant term because we are doing something separate for each of the n customers, and not trying to do something that compares them to each other (for example), so the complexity is $O(n)$.

(c) *Quicksort* works by divide-and-conquer. The input is a list of values to sort. First we pick a pivot value; there are various ways to do this, but the simplest is just to pick the first value in the list. Then we divide the list into three parts: all those items less than the pivot value, the pivot item itself, and all those items greater than or equal to the pivot value. The output is a list composed of the lower part (sorted by calling ourselves recursively), followed by the pivot value, followed by the upper part (again sorted by calling ourselves recursively).

(d) The two defining features of a binary search tree are (1) that, for each node, all the keys in the left subtree are less than the key of the node, while all the keys in the right subtree are greater than the key of the node; and (2) that each key value is present only once.

An insertion algorithm based on these insights. Given the root node of a binary search tree, and a key value: if the node is undefined (ie it's a leaf node), then create a new node with the provided key value and a null left child and a null right child, and return this node; if the node is defined and the key value equals the key of the node then return a value to show the key is already present.

Otherwise if the key value is greater than the value of this node's key, then call ourselves recursively on this node's right child, with the same key, and return the result.

Finally if the key value is less than the value of this node's key, then call ourselves recursively on this node's left child, with the same key, and return the result.

2.20 M269 2013J Exam Soln 17

- See notes version for text

Go to Exam Q 17

M269 Exam 2013J Q 17 Sample Solution

Alan Turing and his Marvellous Machine

Alan Turing who was one of the mathematical and computing heroes of Bletchley Park. Many U3A members may have visited the famous WW2 code-breaking centre at Bletchley Park — some of them may even have worked there — and will know that Turing was closely involved with building the first electronic computers and using them to decipher enemy radio messages. Members may also be interested to know that Turing had been working before the war in Cambridge on the fundamental ideas of computing and can be regarded as one of the founders of modern computer science.

Turing was interested in what humans do when they compute the answer to a problem, and whether this process could be done by a machine, and if it could, what would be be the limits of what such a machine could do. He imagined a simple, idealized machine that could read and write and erase symbols on an endless paper tape, and that could be set up to follow instructions by responding in a particular way to each different symbol. He showed that given enough time and a long enough tape, such a simplified machine could be set up to add up, or take away, or do arbitrarily complex mathematics. These imaginary machines became known as Turing Machines.

Turing's design was a purely theoretical one. But he used it to prove one of the most important practical results about computers: no matter how fast and how efficient we make our computers, there will always be problems that the computer cannot solve. His argument was based on the idea that if you made a long list of all possible Turing Machines that solve a particular type problem, it is always possible to construct another problem that cannot be solved by any possible machine. This fundamental result still shapes the way that computer scientists search for algorithms and develop programs to solve problems.

The real beauty of the Turing Machine is that it is so simple; this simplicity allows computer scientists to reason about computing and what can be computed without the distracting details of any particular real-world machine.