3.36pt

M269 Exams

Prsntn 2015J Exam

Phil Molyneux

24 August 2016

M269 2015J Exam
Q s

- M269 Algorithms, Data Structures and Computability
- Presentation 2015J Exam
- Date Thursday, 2 June 2016 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

M269 2015J Exam
Q Part1

- Answer every question in this part.
- The marks for each question are given below the question number.
- 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.
- Use the provided answer books for any rough working.

Go to Soln Part1

M269 2015J Exam
Q 1

- **Question 1** Which **two** of the following statements are true? **(2 marks)**

A. Computational thinking consists of the skills to formulate a problem as a computational problem and then construct a good computational solution to solve it or explain why there is no such solution.

B. Every computable problem can be solved in a practical way using existing computers.

C. A computational problem is computable if it is possible to build an algorithm which solves every instance of the problem in a finite number of steps.

D. An algorithm consists of a computer program that will solve a computable problem.

Go to Soln 1

M269 2015J Exam
Q 2

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

A. Abstraction allows us to manage complexity.

B. In abstraction as modelling, we hide the details of an implementation behind an interface.

C. Every algorithm can be expressed as some combination of sequence, iteration and selection.

D. If a polynomial algorithm is executed, it will quickly overwhelm the resources of a computer and exceed any reasonable time limits.

Go to Soln 2

M269 2015J Exam

Q 3

- **Question 3** In roughly three or four sentences (in total) explain what is meant by the following terms:

  (4 marks)

- Abstract data type (ADT)
- Encapsulation
- Data structure

Go to Soln 3

M269 2015J Exam

Q 4

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

```python
while (s < 5 and t > 3) or not(s >= 5 or t <= 3):
```

(a) Make the following substitutions:

P represents `s < 5`

Q represents `t > 3`

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						

- Q 4 continued on next slide

Go to Soln 4

M269 2015J Exs Exam

Q 4 (contd)

(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. `not (s < 5 and t > 3)`

B. `(s >= 5 or t <= 3)`

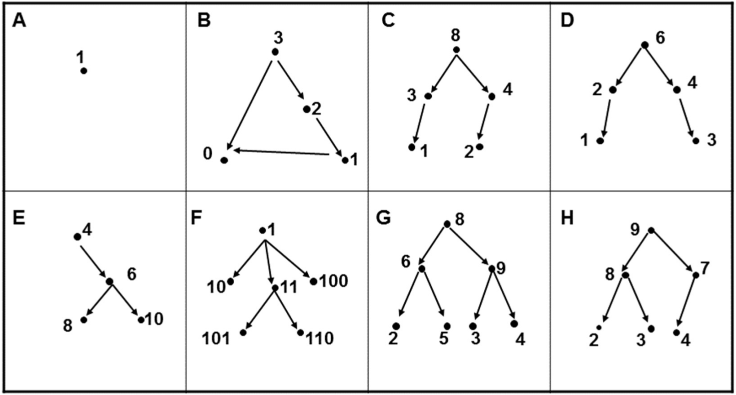
C. `(s < 5 and t > 3)`

D. `(s >= 5 and t <= 3)`

E. `(s < 5 and t <= 3)`

Go to Exam Soln 4

- **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)**



- Q 5 continued on next slide

Go to Soln 5

M269 2015J Exs Exam

Q 5 (contd)

- 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

M269 2015J Exam
Q 6

- **Question 6** Consider the following function, which takes an integer argument `n`. You can assume that `n` is positive. **(4 marks)**

```python
def calculate(n):
    a = 5
    ans = 0
    for i in range(n):
        x = i * i
        for j in range(n):
            y = x + j * j
            for k in range(n):
                z = y + i * k
                ans = ans + z * a
    return ans
```

- Q 6 continued on next slide

▸ Go to Soln 6

M269 2015J Exam

Q 6 (contd)

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

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

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

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

E. $T(n) = 3n + 6$ and $O(n)$

- Now explain how you obtained $T(n)$ and the Big-O complexity.

Go to Exam Soln 6

M269 2015J Exam
Q 7

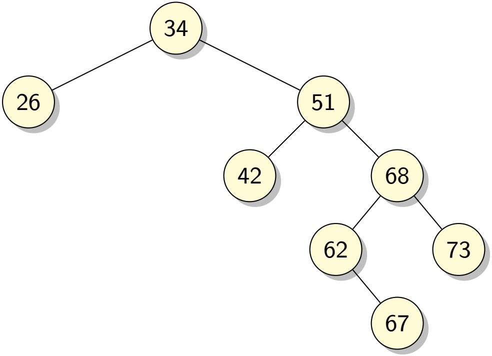
- **Question 7** In the KMP algorithm, for each character in turn, as it appears in the target string T, we identify the longest substring of T ending with that character which matches a prefix of T.
- These lengths are stored in what is known as a prefix table (which in Unit 4 we represented as a list).
- Consider the target string T:
  `CDCECDCECE`
- 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. **(4 marks)**

C	D	C	E	C	D	C	E	C	E
0		1	0		2		4		0

Go to Soln 7

M269 2015J Exam
Q 8

- Consider the following Binary Search Tree. **(4 marks)**



(a) Calculate the balance factors of each node in the above tree and annotate the above tree to show these balance factors.

(b) Redraw the tree after node 51 has been deleted.

Go to Soln 8

M269 2015J Exam
Q 9

- **Question 9** In Python a dictionary of dictionaries can be used to represent a graph's adjacency list. Consider the following: **(4 marks)**

```
graph2 = {
 0:{'neighbours': [1,2,3,4]},
 1:{'neighbours': [0,3,4]},
 2:{'neighbours': [0,5]},
 3:{'neighbours': [0,1,5]},
 4:{'neighbours': [0,1]},
 5:{'neighbours': [2,3]}}
```

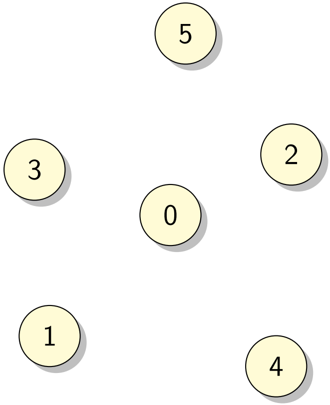
- Q 9 continued on next slide

Go to Soln 9

M269 2015J Exam

Q 9 (contd)

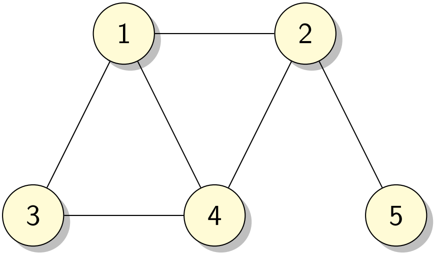
- In the space provided below, **complete** the graph corresponding to the adjacency list given above.



Go to Exam Soln 9

M269 2015J Exam
Q 10

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



- In the space provided below, draw **one** spanning tree that could be generated from a **Breadth First Search** of the above graph starting at vertex 2.

Go to Soln 10

M269 2015J Exam
Q 11

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

(a) What does it mean to say that two well-formed formulas (WFFs) are **logically equivalent** ? Use the space below for your answer.

(b) Is the following set of propositional WFFs **satisfiable** ?

$$\{(P \rightarrow Q), (Q \rightarrow P)\}$$

- Explain how you arrived at your answer in the space below:

Go to Soln 11

M269 2015J Exam
Q 12

- **Question 12** Consider a domain with some board games and people. **(6 marks)**

  $\mathcal{D} = \{\text{Backgammon, Chess, Draughts, Joe, Mary, Sue}\}$
- An interpretation assigns people to corresponding constants (you won't need the constants for games).

  $\mathcal{I}(joe) = \text{Joe}$
  $\mathcal{I}(mary) = \text{Mary}$
  $\mathcal{I}(sue) = \text{Sue}$
- The predicates *owns* and *likes* are assigned to binary relations with the following comprehensions:

  $\mathcal{I}(owns) = \{(P, G)\text{: the person } P \text{ owns the game } G\}$

  $\mathcal{I}(likes) = \{(P, G)\text{: the person } P \text{ likes the game } G\}$
- Q 12 continued on next slide

Go to Soln 12

M269 2015J Exam

Q 12 (contd)

- The enumerations of the relations are:
- $\mathcal{I}(owns) = \{$(Joe, Chess), (Mary, Backgammon), (Sue, Draughts)$\}$
- $\mathcal{I}(likes) = \{$(Joe, Backgammon), (Mary, Backgammon), (Mary, Draughts), (Sue, Backgammon), (Sue, Chess)$\}$
- You will find the questions on the next page.
- You are asked to translate a sentence of predicate logic to English or vice-versa.
- 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.

- Q 12 continued on next slide

Go to Exam Soln 12

M269 2015J Exam

Q 12 (contd)

- In your explanation you need to consider any relevant values for the variables, and *show, using the interpretation above, whether they make the quantified expression TRUE or FALSE*.
- When your explanation refers to the interpretation, make sure that you use formal notation.
- So instead of saying that *Joe likes Backgammon according to the interpretation*, write:
  (Joe, Backgammon) $\in \mathcal{I}(\textit{likes})$.
- Similarly, instead of *Joe doesn't like Backgammon* you would need to write:
  (Joe, Backgammon) $\notin \mathcal{I}(\textit{likes})$.

- Q 12 continued on next slide

Go to Exam Soln 12

M269 2015J Exam
Q 12 (contd)

(a) $\forall X.(owns(joe, X) \rightarrow likes(joe, X))$

can be translated into English as:

- This sentence is ____ (choose from TRUE/FALSE), because:

(b) *There's something that both Mary and Sue like*

can be translated into predicate logic as:

- This sentence is ____ (choose from TRUE/FALSE), because:

Go to Exam Soln 12

- **Question 13** The interpretation of the previous question can also be represented by a database with the following tables, *owns* and *likes*. **(6 marks)**

owns	
owner	*boardgame*
Joe	Chess
Mary	Backgammon
Sue	Draughts

likes	
person	*game*
Joe	Backgammon
Mary	Backgammon
Mary	Draughts
Sue	Backgammon
Sue	Chess

- Q 13 continued on next slide

Go to Soln 13

M269 2015J Exam

Q 13 (contd)

(a) For the following SQL query, give the table returned by the query.

```sql
SELECT person
FROM owns CROSS JOIN likes
WHERE game = boardgame AND person = owner;
```

- Write the question that the above query is answering.

(b) Write an SQL query that answers the question *Which games does Sue like?*

- The answer should be the following table:

game
Backgammon
Chess

Go to Exam Soln 13

- **Question 14** Consider the following axiom schema and rules: **(4 marks)**
- *Axiom schema* $\{\boldsymbol{A}, \boldsymbol{B}\} \vdash \boldsymbol{A}$
- *Rules*
- $$\frac{\boldsymbol{\Gamma} \vdash \boldsymbol{A} \wedge \boldsymbol{B}}{\boldsymbol{\Gamma} \vdash \boldsymbol{A}} \ (\wedge\text{-elimination left})$$
- $$\frac{\boldsymbol{\Gamma} \vdash \boldsymbol{A} \wedge \boldsymbol{B}}{\boldsymbol{\Gamma} \vdash \boldsymbol{B}} \ (\wedge\text{-elimination right})$$
- $$\frac{\boldsymbol{\Gamma} \vdash \boldsymbol{A} \quad \boldsymbol{\Gamma} \vdash \boldsymbol{B}}{\boldsymbol{\Gamma} \vdash \boldsymbol{A} \wedge \boldsymbol{B}} \ (\wedge\text{-introduction})$$
- $$\frac{\boldsymbol{\Gamma} \cup \{\boldsymbol{A}\} \vdash \boldsymbol{B}}{\boldsymbol{\Gamma} \vdash \boldsymbol{A} \rightarrow \boldsymbol{B}} \ (\rightarrow\text{-introduction})$$
- $$\frac{\boldsymbol{\Gamma} \vdash \boldsymbol{A} \quad \boldsymbol{\Gamma} \vdash \boldsymbol{A} \rightarrow \boldsymbol{B}}{\boldsymbol{\Gamma} \vdash \boldsymbol{B}} \ (\rightarrow\text{-elimination})$$
- Q 14 continued on next slide

Go to Soln 14

M269 2015J Exam

Q 14

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

1.	$\{V, W\} \vdash V$	[Axiom schema]
2.	$\boxed{??\qquad\qquad ??}$	[Axiom schema]
3.	$\{V, W\} \vdash V \land W$	$\boxed{??\qquad\qquad ??}$

Go to Exam Soln 14

M269 2015J Exam
Q 15

- **Question 15** Which **two** of the following statements are true? (Tick **two** boxes.)

(a) The Halting Problem is semi-decidable.

(b) The Equivalence Problem is computable.

(c) The Church-Turing Thesis proves that all definitions of an algorithm are equivalent.

(d) A reduction from a non-computable problem A to a problem B proves that B is not computable.

- Note that the original exam did not have labels for the boxes

Go to Soln 15

M269 2015J Exam

Q Part2

- Answer every question in this Part.
- The marks for each question are given below the question number.
- Marks for a part of a question are given after the question.
- Answers to questions in this Part must be written in the additional answer books, which you should also use for your rough working.

Go to Soln Part2

M269 2015J Exam

Q 16 *(25 marks)*

- The Universal Product Corporation (UPC) keeps rather primitive computerised records of its sales of a range of world class products.
- These are contained in a sequence S of sales, where each sale records the number sold of a particular product, in the form of `[productCode, numberSold]`.
- The sequence S lists the sales as they were processed, from first to last.
- The sequence has at least one sale. Each product has a different `productCode`. There may be multiple sales for the same product.
- Q 16 continued on next slide

▸ Go to Soln 16

M269 2015J Exam

Q 16 (contd)

- An example sequence S is, in Python notation:

  `[['PR1', 5], ['B20', 10], ['PR1', 3]]`

(a) The company requires a function that returns a sequence of how many sales were processed for each product. For example, the example sequence S given above would lead to an output of either:

`[['PR1', 2], ['B20', 1]]` or

`[['B20', 1], ['PR1', 2]]`

- showing that there are two sales for product `'PR1'` and one for product `'B20'`.

- Q 16 continued on next slide

Go to Soln 16

M269 2015J Exam
Q 16 (contd)

- Using the following template, formally state this as a computational problem, in the style adopted by M269. **(6 marks)**

  Name: SalesSummary
  Inputs:
  Preconditions: *(indicate only* ***one****)*
  Outputs:
  Postconditions: *(indicate only* ***one****)*

- Q 16 continued on next slide

Go to Soln 16

M269 2015J Exam
Q 16 (contd)

(b) UPC want a function that returns the code of the product with the fewest sales processed, so that UPC can start promoting it.

- If the lowest number of sales is shared by several products, the function can return the product code of any one of them.
- A UPC employee has the following initial insight:
  - *Take the sales summary sequence (i.e. the output of SalesSummary) and use QuickSort to sort it in ascending order by the number of sales.*
  - *This will put one of the products with fewest sales in the first position, so then just return the product code of the first element of the sorted sequence.*

- Q 16 continued on next slide

Go to Soln 16

M269 2015J Exam
Q 16 (contd)

(i) What is the order of complexity, in Big-O notation, of the algorithm described by the employee's initial insight, in the best case?

- Assume that SalesSummary has already run.

(ii) Give the initial insight of a more efficient solution and state its order of complexity in Big-O notation.

(6 marks)

- Q 16 continued on next slide

Go to Soln 16

M269 2015J Exam
Q 16 (contd)

(c) UPC introduce a further data sequence P, which is an unsorted sequence of product prices, such that each item in the sequence is in the form of `[productCode, price]` and each product is included exactly once.

(i) A function is required that will return the total value of all sales for each product.

- So given the sequence S of sales, each in the form `[productCode, numberSold]`:
  `[['PR1', 5], ['B20', 10], ['PR1', 3]]`
- the output of the function would be:
  `[['B20', 49.9], ['PR1', 28.0]]`
  or `[['PR1', 27.5], ['B20', 49.9]]`
- This is because 10 items of product `'B20'` ($10 \times 4.99 = 49.9$) and 8 items of product `'PR1'` ($8 \times 3.50 = 28.0$) were sold.
- Write structured English or Python code for a computational solution of this problem.
- Q 16 continued on next slide

Go to Soln 16

M269 2015J Exam

Q 16 (contd)

(ii) Estimate the run time T() and the order of complexity in Big-O notation of your solution, in the worst case, taking assignment as the unit of computation.

- Explain your reasoning. If you make any assumptions, state them clearly. **(9 marks)**

- Q 16 continued on next slide

Go to Soln 16

M269 2015J Exam

Q 16 (contd)

(d) Storing complex data items (e.g. product/price combinations) in a list, as UPC have opted to do, can be problematic, in particular because retrieval may be slow when there are very large numbers of items. A more suitable means of storage is in a structure such as a *hash table*.

- Write roughly four to six sentences explaining: **(4 marks)**
  - the basic principles of *hash tables*
  - and hashing,
  - how these could work with the UPC price data, and
  - outline one problem that can arise from hashing.

Go to Soln 16

M269 2015J Exam

Q 17 *(15marks)*

- Your local secondary school runs a computer club for sixth form students.
- You have been asked to give a talk on *greedy algorithms* and, in preparation, to prepare a report for the teachers summarising your talk.
- Assume that the students and teachers do not have a background in computer science, but have been writing programs in various computer languages and are IT literate.
- Q 17 continued on next slide

Go to Soln 17

M269 2015J Exam

Q 17 (contd)

- Your report **must** have the following structure:

1. A suitable title
2. A paragraph *setting the scene*: explain in layperson's terms what is meant by *a greedy algorithm* and give an example of where greed is not always good.
3. A paragraph in which you describe a minimum spanning tree (MST) and give an example of one. You don't need to explain what are trees and graphs.
4. A paragraph in which you briefly describe what is Prim's algorithm and some of its features. You do not need to describe Prim's algorithm completely.
5. A concluding paragraph, giving reasons, about the benefits or otherwise of a greedy algorithm.

- Q 17 continued on next slide

Go to Soln 17

M269 2015J Exam
Q 17 (contd)

- 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.
- As a guide, you should aim to write roughly three to five sentences per paragraph.

Go to Soln 17

M269 2015J 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

M269 2015J Exam
Soln Part1

- Part 1 solutions

Go to Q Part1

M269 2015J Exam

Soln 1

- A, C

Go to Q 1

M269 2015J Exam

Soln 2

- A, C

Go to Q 2

M269 2015J Exam
Soln 3

- An *abstract data type* is a logical description of how we view the data and the operations that are allowed without regard to how they will be implemented. See Miller and Ranum chp 1 and Wikipedia: Abstract data type
- *Encapsulation* hides the implementation of an ADT so a user must only access data via the interface and not directly.
- A *data structure* is a concrete implementation of some ADT

Go to Q 3

M269 2015J Exam

Soln 4

(a) 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	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) C

`(s < 5 and t > 3)` is equivalent to

```
(s < 5 and t > 3) or not(s >= 5 or t <= 3)
```

Go to Q 4

M269 2015J Exam

Soln 5

(a) B is not a tree; it has more than one route from node 3 to node 0.

(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 H is a heap; (complete binary tree, and parent nodes $>$ children).

Go to Q 5

M269 2015J Exam

Soln 6

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

- 2 assignment statements outside the loops
- 1 assignment statement in the outer loop
- 1 assignment statement in the middle loop
- 2 assignment statements in the inner loop
- n^3 is the dominant term

Go to Q 6

M269 2015J Exam

Soln 7

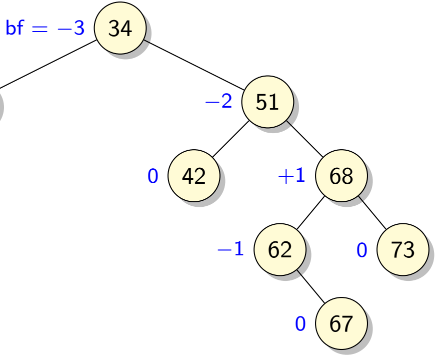
C	D	C	E	C	D	C	E	C	E
0	0	1	0	1	2	3	4	1	0

Go to Q 7

M269 2015J Exam

Soln 8

(a) Balance factors



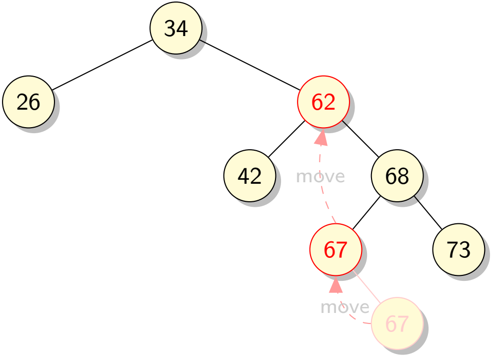
- Soln 8 continued on next slide

Go to Q 8

M269 2015J Exs Exam

Soln 8 (b)

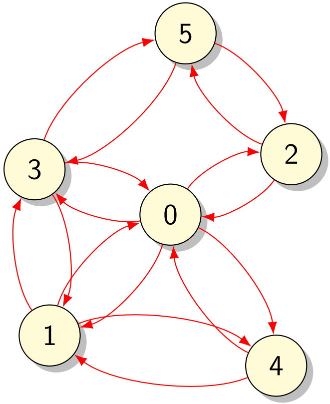
(b) Delete 51



Go to Exam Q 8

M269 2015J Exam
Soln 9

- Here is a representation with unidirectional edges

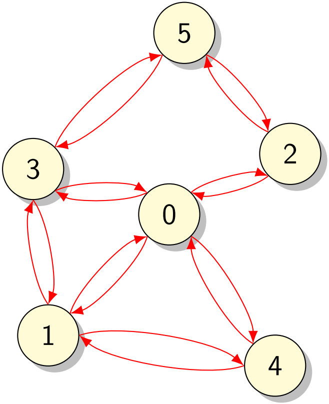


- Soln 9 continued on next slide

Go to Q 9

M269 2015J Exam
Soln 9 (contd)

- Here is a representation with unidirectional edges

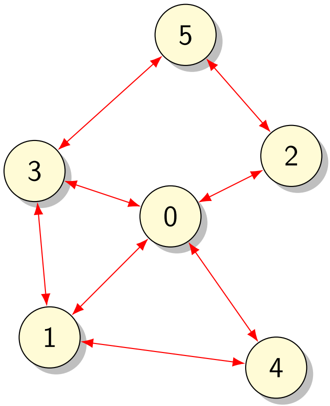


- Soln 9 continued on next slide

Go to Exam Q 9

M269 2015J Exs Exam
Soln 9 (contd)

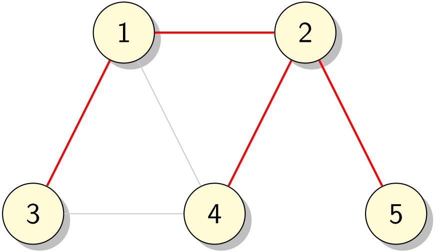
- Here is a representation with bidirectional edges (but we have not been told that every edge has a reverse edge and of the same weight or length)



Go to Exam Q 9

M269 2015J Exam
Soln 10

- *Spanning tree* from *breadth first search* from vertex 2 (1 of 2, in red)



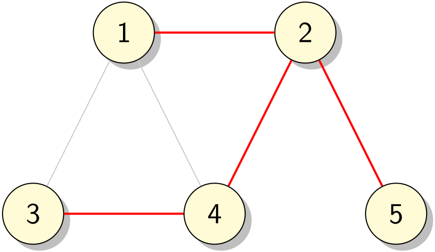
- Soln 10 continued on next slide

Go to Q 10

M269 2015J Exs Exam

Soln 10

- *Spanning tree* from *breadth first search* from vertex 2 (2 of 2, in red)



Go to Exam Q 10

M269 2015J Exam

Soln 11

(a) Two well-formed formulas (WFFs) **A** and **B** are logically equivalent if and only if **A** and **B** have the same value in all interpretations.

(b) The sets of WFFs is *satisfiable* if each member has the value *True* for some interpretation

P	Q	$P \to Q$	$Q \to P$
T	T	T	T
T	F	F	T
F	T	T	F
F	F	T	T

- The set is satisfiable

Go to Q 11

M269 2015J Exam
Soln 12

(a) Joe likes the games he owns

- False — Joe owns Chess but does not like it
- (Joe, Chess) $\in \mathcal{I}(\mathit{owns})$
- but (Joe, Chess) $\notin \mathcal{I}(\mathit{likes})$

(b) $\exists X.(\mathit{likes}(\mathit{mary}, X) \land \mathit{likes}(\mathit{sue}, X))$

- True — they both like Backgammon

Go to Q 12

M269 2015J Exam
Soln 13

(a) The table

Mary
Sue

- Who owns games they like ?

(b) The SQL query

```
SELECT game
FROM likes
WHERE person = 'Sue';
```

Go to Q 13

M269 2015J Exam

Soln 14

- Completed proof

1.	$\{V, W\} \vdash V$	[Axiom schema]
2.	$\boxed{\{V, W\} \vdash W}$	[Axiom schema]
3.	$\{V, W\} \vdash V \wedge W$	$\boxed{[\wedge\text{-introduction}]}$

Go to Q 14

M269 2015J Exam

Soln 15

- **Question 15** Which **two** of the following statements are true? (Tick **two** boxes.)

(a) The Halting Problem is semi-decidable. **True** See Wikipedia: Decidability (logic)

(b) The Equivalence Problem is computable. **False**

(c) The Church-Turing Thesis proves that all definitions of an algorithm are equivalent. **False**

(d) A reduction from a non-computable problem A to a problem B proves that B is not computable. **True**

Go to Q 15

M269 2015J Exam

Soln Part2

- Part 2 solutions

Go to Q Part2

M269 2015J Exam

Soln 16

(a) **Name:** SalesSummary

Inputs: An unsorted sequence of tuples $S = (s_1, s_2, \ldots, s_n)$ where $s_n = (p_n, q_n)$ and `productCode`, p_n is a string, and `numberSold`, q_n, is an integer.

Preconditions: length $S \geqslant 1$

Outputs: a list of tuples $O = (o_1, o_2, \ldots, o_m)$ where $o_p = (p_m, r_m)$ and p_m is a `productCode` and r_m is an integer.

Postconditions: Length O equals number of product codes

- Soln 16 continued on next slide

Go to Q 16

M269 2015J Exam

Soln 16 (contd)

(b) **(i)** Complexity of Quicksort in the best case is $O(n \log n)$ and worst case is $O(n^2)$ (see Big-O Cheat Sheet)

- **(ii)** A linear search can find the smallest — $O(n)$

- Soln 16 continued on next slide

Go to Q 16

M269 2015J Exam

Soln 16 (contd)

(c) **(i)** Sketch of programming strategy

- Sort the sequence of sales using Python's `Timsort` — worst case complexity $O(n \log n)$
- Group tuples for each product into sub-sequences — one traversal of the sequence $O(n)$
- For each sub-sequence calculate the value of a sale and sum the values — one traversal of each sub-sequence $O(n)$
- **(ii)** Overall complexity $O(n \log n)$

- Soln 16 continued on next slide

Go to Q 16

M269 2015J Exam

Soln 16 (contd)

(d) Hash function and hash tables

- Hash function maps each input key to a hash value (or slot)
- Perfect hash function maps each key to a different hash value
- For UPC could translate `productCode` to an integer by using Unicode or ASCII values for each character
- Limited storage leads to hash functions having collisions — a hash function mapping two keys to the same slot
- Hash function collisions result in the need to either store multiple items in a single slot (closed table) or open addressing/open tables that use some mechanism to find a free slot

Go to Q 16

M269 2015J Exam
Soln 17

- **Title** Greed is (sometimes) good
- Define **Graph** and **Tree** with example
- Define **Minimum Spanning Tree** of a graph is the spanning tree (includes every node but may not include every edge) that minimises total weight of edges
- Describe **Prim's algorithm** — repeatedly add the next *safe* edge — the only safe edge will be the one with the smallest edge from the tree so far
- Greed is (hardly ever) good — give an example where it does not work — knapsack problem.

Go to Q 17