UNIVERSITY OF SOUTHAMPTON COMP1201W1

SEMESTER 2 EXAMINATION 2014/2015

ALGORITHMICS

Duration: 120 mins

You must enter your Student ID and your ISS login ID (as a cross-check) on this page. You must not write your name anywhere on the paper.

Student ID:	
ISS ID:	

Question	Marks
1	
2	
3	
4	
Total	

Answer all parts of the question in section A (25 marks) and TWO *questions from section B (30 marks each).*

This examination is worth 85%. The tutorials were worth 15%.

University approved calculators MAY be used.

A foreign language translation dictionary (paper version) is permitted provided it contains no notes, additions or annotations.

Each answer must be completely contained within the box under the corresponding question. No credit will be given for answers presented elsewhere.

You are advised to write using a soft pencil so that you may readily correct mistakes with an eraser.

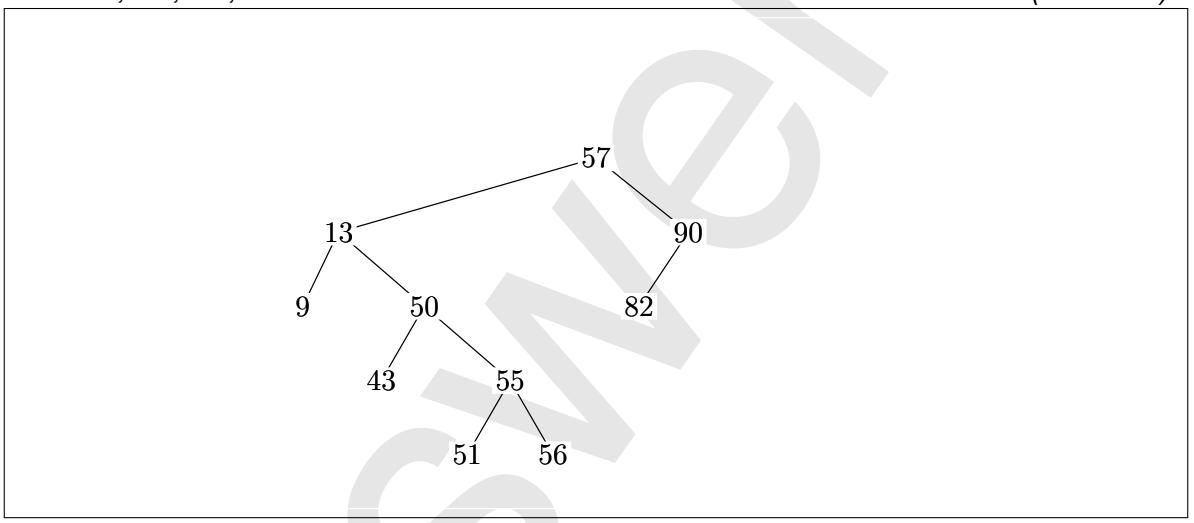
You may use a blue book for scratch—it will be discarded without being looked at.

 Number of pages 14

Section A

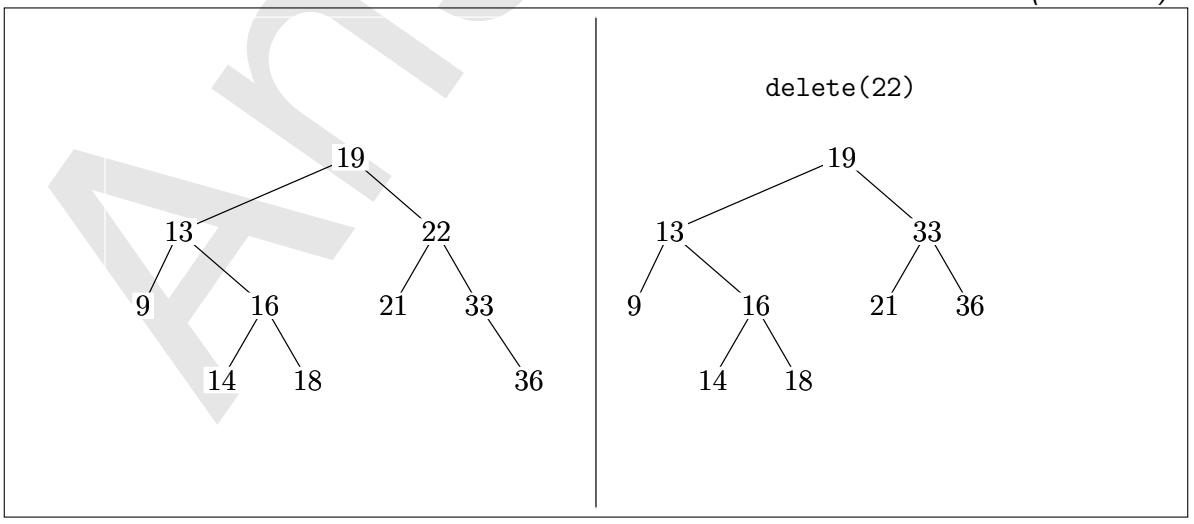
Question A 1

(a) Draw the binary search tree produced when inserting 57, 90, 13, 9, 50, 43, 55, 82, 56, 51. *(2 marks)*



2

(b) Draw the tree obtained by deleting 22 from the binary search tree shown. *(2 marks)*



2

(c) What type of binary search tree is used in the java TreeSet? *(1 marks)*

Red-black trees

1

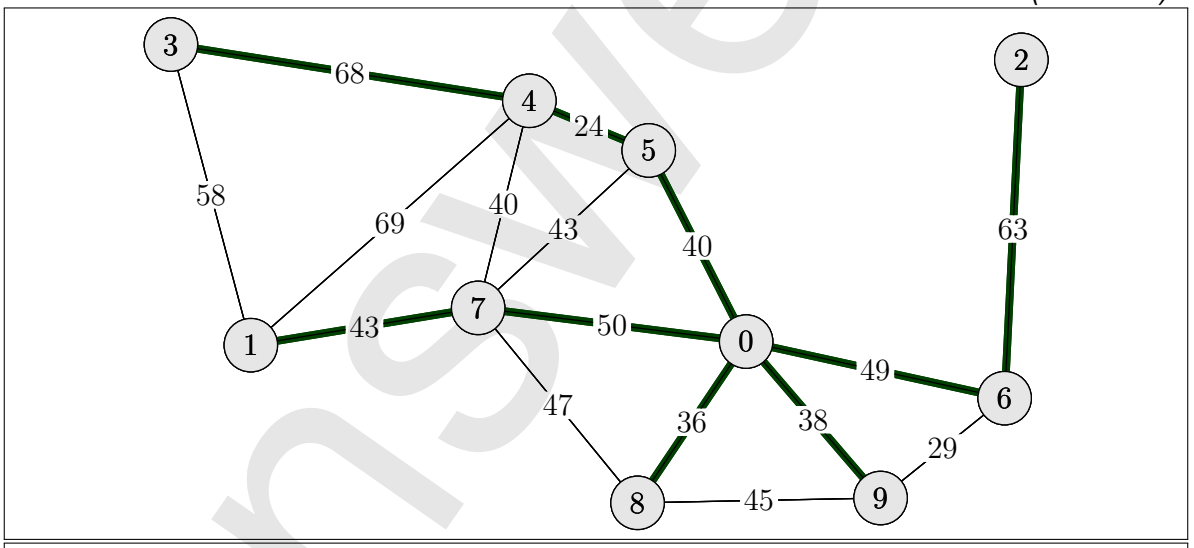
(d) When would you prefer to use a set rather than a list and the vice versa? *(3 marks)*

3

Sets are used when you wish to quickly check that an element is contained in the list. A list is necessary if you care about the order in which data is entered and if data is repeated (although multisets will cope with repetition).

(e) Show the tree of edges found by Dijkstra's algorithm from node 0 and write down the order of the edges and the distance of the node to the source node. *(7 marks)*

7

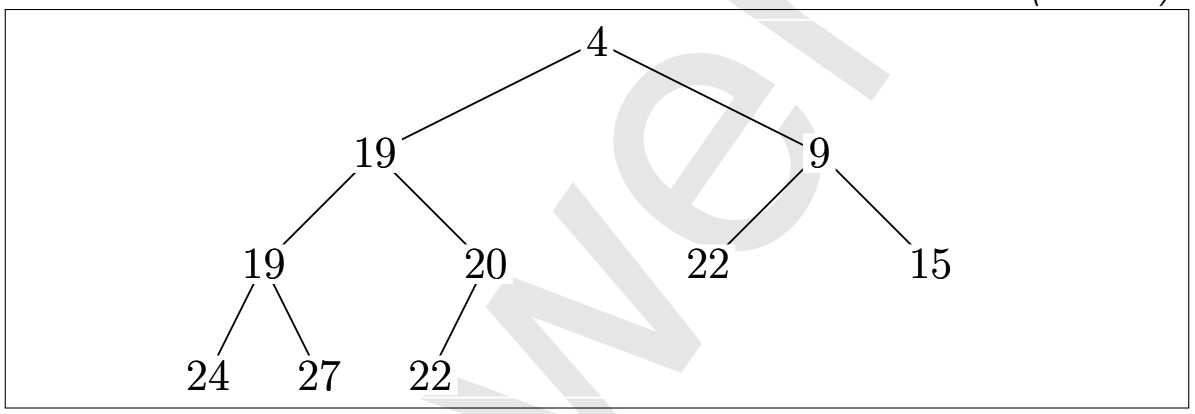


1. (0,8) 36	2. (0,9) 38	3. (0,5) 40
4. (0,6) 49	5. (0,7) 50	6. (4,5) 64
7. (1,7) 93	8. (2,6) 112	9. (3,4) 132

(f) Heaps use a binary tree encoded into an array. Show the binary tree represented by the following array.

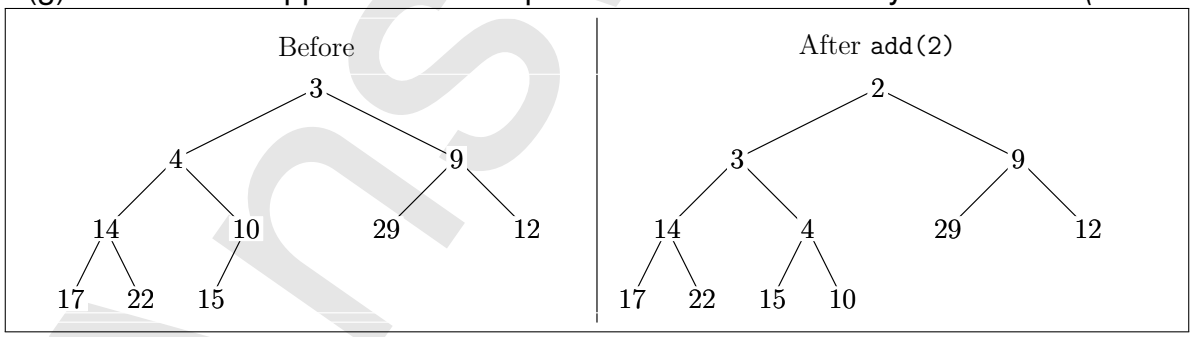
4	19	9	19	20	22	15	24	27	22

(1 marks)



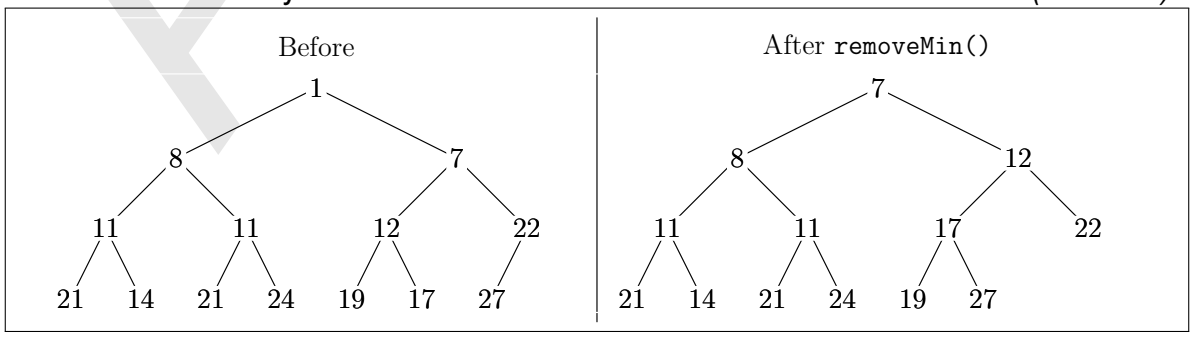
1

(g) Show what happens to the heap shown on the left when you add 2. *(2 marks)*



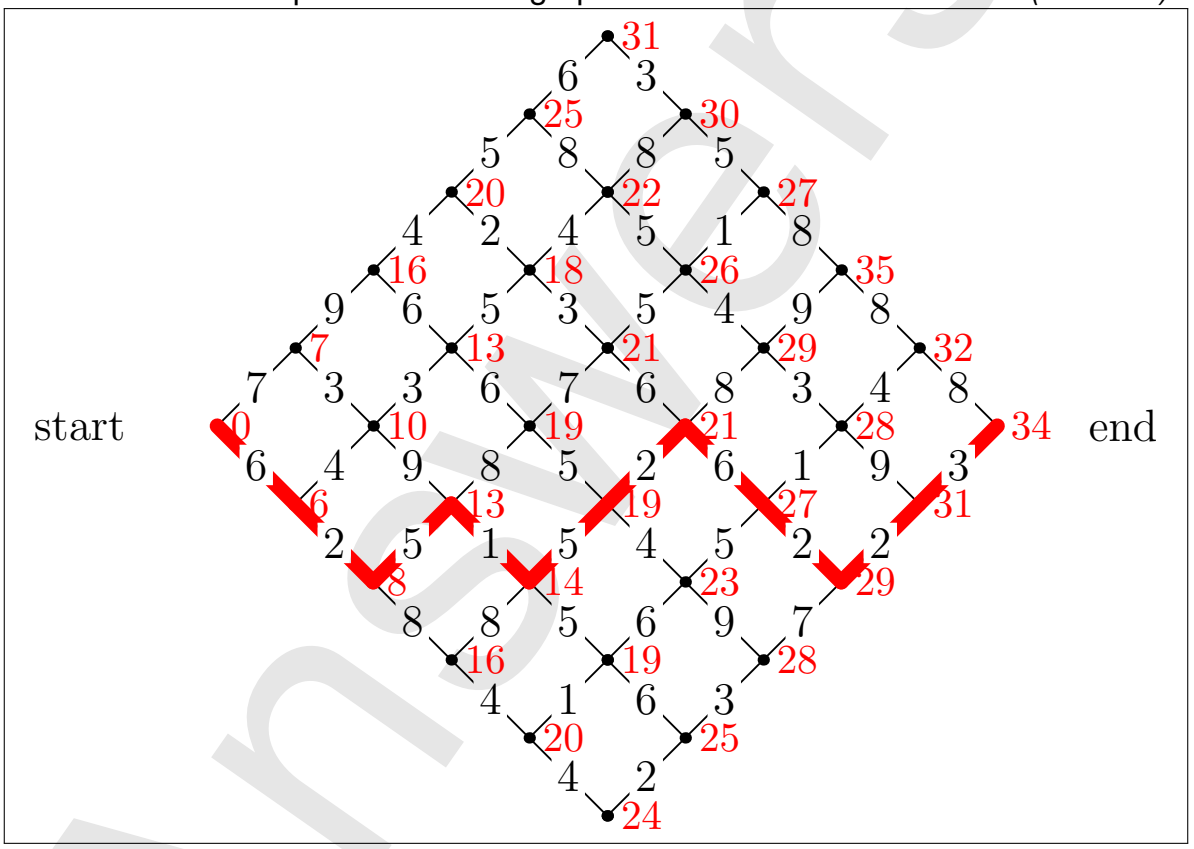
2

(h) Show what happens to the heap shown on the left when you remove the minimum entry. *(2 marks)*



2

(i) Use the dynamic programming forward algorithm to compute the minimum cost of each path from the left most node to each other node where the cost of moving along an edge is equal to the number shown. An edge can only be traversed from left to right. Use the backwards algorithm to find the minimum cost path across the graph. *(5 marks)*



5

End of question 1

Q1:	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	Total
	/2	/2	/1	/3	/7	/1	/2	/2	/5	/25

Section B

Question B 2

(a) What is the main application of B-Trees? *(1 marks)*

Databases

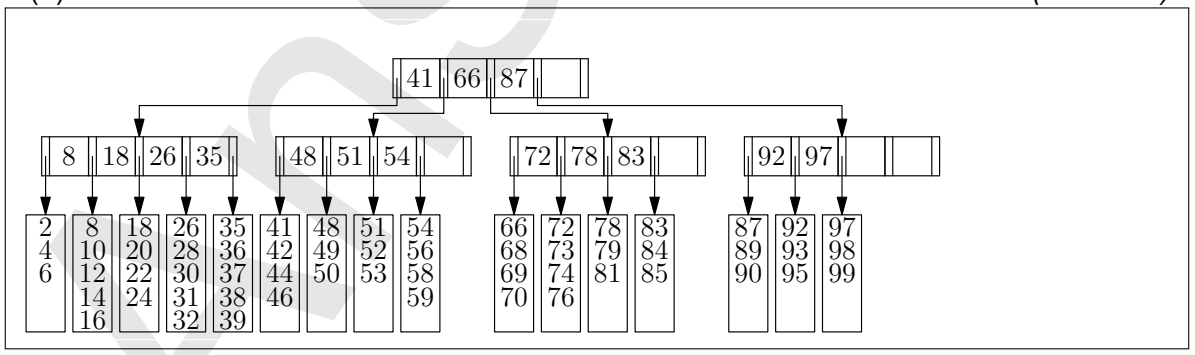
1

(b) Describe what problem B-Trees solve and how they solve it. *(3 marks)*

B-Trees are trying to minimise the number of data access on secondary storage, such as disks, as this is much slower (by up to 10^7) than the CPU clock time. It uses a multi-way tree to minimise the number of data lookups to reach the root of a tree.

3

(c) Sketch a B-Tree. *(6 marks)*



6

(d) What is a trie or digital tree *(3 marks)*

A trie is a multi-way tree for storing words composed of letters from a restricted alphabet. The split is done on each letter of the alphabet. The tree is collapsed when there are no further splits.

3

(e) What is (1) the advantage and (2) disadvantage of using a trie compared with a binary search tree? *(2 marks)*

(i) Tries offer faster operations because the depth of the tree is much shallower.

(ii) Tries use up much more memory as they are usually sparsely occupied.

2

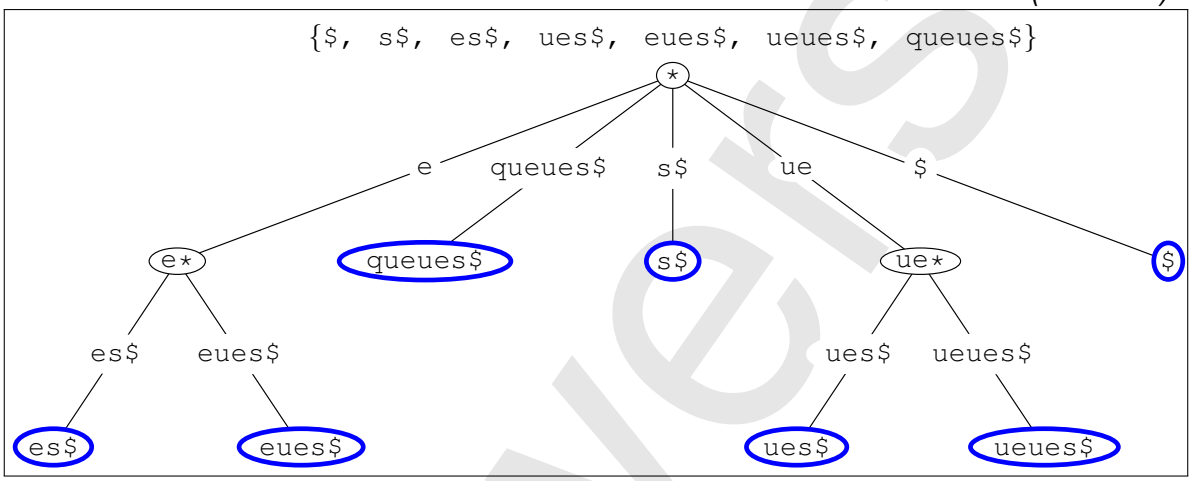
(f) Show how the following words would be inserted into a trie based on tables *(10 marks)*

Add words: {BIN, BINARY, MAP, MAPPER, MAPPERS, MAPS, OTHER, SET, SETTER, THAT, THE, THIS, TREE}

	0	1	2	3	4	5	6
$		BIN$	MAP$	MAPPER$	SET$		
A		BINARY$				THAT$	
B	BIN* (1)						
C							
D							
E						THE$	TH* (5)
I						THIS$	
L							
M	MAP* (2)						
N							
O	OTHER$						
P			MAPPER* (3)				
R							TREE$
S	SET* (4)		MAPS$	MAPPERS$			
T	T* (6)				SETTER$		
U							

10

(g) Write down all the suffixes of the word "**queues**" and draw the suffix tree for the word. *(5 marks)*



End of question 2

5

Q2: (a) —	(b) —	(c) —	(d) —	(e) —	(f) —	(g) —	Total —
	1	3	6	3	2	10	5	30

Question B 3

The quicksort algorithm is as follows

```
QUICKSORT(a, left, right) {
  if (right-left < threshold)
    INSERTIONSORT(a, left, right)
  else
    pivot = CHOOSEPIVOT(a, left, right)
    part = PARTITION(a, pivot, left, right)
    QUICKSORT(a, left, part)
    QUICKSORT(a, part+1, right)
  endif
}
```

(a) Describe the CHOOSEPIVOT algorithm for finding a pivot. *(3 marks)*

(Test understanding of qsort.)

A reasonable algorithm is to choose the median of the first, middle and last element of the array.

3

(b) How does the PARTITION algorithm work? *(5 marks)*

The algorithm works by starting from either side of the array. It finds the first element on the left side that is greater than the pivot and finds the first element on right side that is less than or equal to the pivot. The two elements are swapped. This is repeated until the two searches meet.

(c) Explain why quicksort uses INSERTIONSORT? *(3 marks)*

For small array sizes insertion sort is faster than quicksort.

3

(d) Assume that PARTITION takes n operations and that the pivot splits the array exactly in half at each step (assume also that the `threshold` equals 1). Write a recursion relations for the number of partitioning operations, $T(n)$. *(4 marks)*

(Test knowledge of computing time complexity.)

4

$$T(n) = 2T(n/2) + n$$

(e) Show that $T(n) = n \log_2(n)$ satisfies the recursion relation in part (d). *(4 marks)*

$$\begin{aligned} T(n) &= 2T(n/2) + n \\ &= 2\left(\frac{n}{2}\log_2\left(\frac{n}{2}\right)\right) + n \\ &= n\log_2\left(\frac{n}{2}\right) + n \\ &= n\left(\log_2(n) - \log_2(2)\right) + n = n\log_2(n) \end{aligned}$$

4

(f) Assume that PARTITION takes n operations and that the pivot splits the array into an array of size $n-1$ and another array of size 1. Write a recursion relations for the number of partitioning operations, $T(n)$. *(4 marks)*

$$T(n) = T(n-1) + T(1) + n$$

4

(g) Show that if $T(1) = 0$ then the time complexity, $T(n)$, of quicksort (given the unlucky partitioning described in part (f)) is equal to the function $f(n) = n(n+1)/2 - 1$. *(4 marks)*

This is a simple proof by induction. The base case follows since $f(1) = 0 = T(1)$.

Assuming $T(n-1) = f(n-1)$ then

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

$\frac{}{4}$

(h) Give the worst case complexity for quicksort and explain why this does not put off people using it. *(3 marks)*

The worst case complexity of quicksort is $\Theta(n^2)$. However, this only happens if we have a very unlucky partitioning. By choosing a pivot to be the median of the first, middle and last partition we make choosing a bad pivot consistently to be very unlikely.

End of question 3

$\frac{}{3}$

Q3: (a) $\frac{}{3}$	(b) $\frac{}{5}$	(c) $\frac{}{3}$	(d) $\frac{}{4}$	(e) $\frac{}{4}$	(f) $\frac{}{4}$	(g) $\frac{}{4}$	(h) $\frac{}{3}$	Total $\frac{}{30}$

• Do not write in this space •

Question B 4

(a) Which of the following algorithms are in class P and which are in class NP-complete *(5 marks)*

(i) TSP: NP-complete

(ii) Minimum Spanning Tree: P

(iii) Maximum Flow: P

(iv) Graph Colouring: NP-complete

(v) Linear Assignment: P

5

(b) Describe what it means to be class NP-complete. *(5 marks)*

NP-complete problem are a class of decisions problems in NP which can be used to solve any other NP-complete problem through a polynomial time reduction. Thus they are the hardest problems in the class of NP problems. There are no known polynomial time algorithms for any NP-complete problems.

5

(c) Describe neighbourhood search and explain how it could be used to find a good solution to an optimisation problem. *(5 marks)*

Neighbourhood search starts from some random configuration and looks at a neighbour (or many neighbours). It will move to the neighbour if the solution is better (has lower cost), or, at least, does not have a worst cost. This is iterated either until a local optimum is reached or we have reached some time limit.

(d) Briefly describe simulated annealing and why it is used. *(5 marks)*

5

Simulated annealing modifies neighbourhood search by sometimes moving in the wrong direction (i.e. to high cost solutions). The probability of making a wrong move decreases over time. It is used to prevent the search from being trapped in local optima.

5

(e) Briefly describe branch and bound and its expected performance. *(5 marks)*

Branch and bound systemically searches a tree of all possible solutions, building up each solution from partial solutions. It stops the search down branches of the search tree when it knows that no solution in the branch can be better than its current best solution. Branch and bound will find the optimum, but can take exponential time, albeit much quicker than exhaustive search.

5

(f) Briefly describe how dynamic programming can be used to solve TSP and describe its time complexity *(5 marks)*

Dynamic programming can be used to solve TSP by finding the optimal paths through each subset of cities starting from subsets of size 1 (for each subset we have to calculate the optimal path entering and leaving from every possible pair of cities). Enlarging the subsets is fast, but there are $\Theta(2^n)$ subsets so the time complexity is $n\,2^n$. This is hopelessly slow to be used in practice, but is actually the best known algorithm.

5

End of question 4

Q4: (a) $\frac{}{5}$ (b) $\frac{}{5}$ (c) $\frac{}{5}$ (d) $\frac{}{5}$ (e) $\frac{}{5}$ (f) $\frac{}{5}$ Total $\frac{}{30}$