# The University of Nottingham

SCHOOL OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

A LEVEL B MODULE, AUTUMN SEMESTER 2001-2002

**ALGORITHMS AND DATA STRUCTURES**

Time allowed TWO Hours

*Candidates must NOT start writing their answers until told to do so*

***Answer QUESTION ONE and THREE other questions***

*No calculators are permitted in this examination.*

*Dictionaries are not allowed with one exception. Those whose first language is not English may use a dictionary to translate between that language and English provided that neither language is the subject of this examination. No electronic devices capable of storing and retrieving text may be used.*

1. **This question is compulsory.** In each part (a)-(e) of the question there are five choices (i)-(v). For each of these choices you should state whether it is true or false. For each correct answer you get one point. For each incorrect answer you get one point subtracted. However you can never get less than 0 points for each part.

   (a) The smallest class to which the function $S(N) = log_2(N) \times (N+1)$ belongs is: (5)

   i. $O(N^2)$
   ii. $O(logN)$
   iii. $O(NlogN)$
   iv. $O(N^2logN)$
   v. $O(N)$

   (b) The following data structure has O(log N) insertion procedure in the worst case (5)

   i. ordered linked list
   ii. ordered array
   iii. binary search tree
   iv. balanced binary search tree
   v. heap

   (c) The following property is an invariant of the loop:

```
for(int i = 0; i < arr.length; i++){
   if (arr[i] == value) return true;
}
```

(5)

   i. for all `k` such that `0` $\leq$ `k < i`, `arr[k]` does not equal `value`.
   ii. for all `k` such that `0` $\leq$ `k` $\leq$ `i`, `arr[k]` does not equal `value`.
   iii. `arr[i]` does not equal `value`.
   iv. `i` $\geq$ `0`.
   v. `arr[i]` equals `value`.

*Continued on next page*

(d) Time complexity of breadth-first search ($V$ is the number of vertices, $E$ the number of edges) is in (5)

i. $O(V)$

ii. $O(V + E)$

iii. $O(E)$

iv. $O(V^2)$

v. $O(E^2)$

(e) The following data structure minimises disk reads when searching for an item of data stored in external memory (5)

i. linked list

ii. B-tree

iii. binary search tree

iv. balanced binary search tree

v. hash table

2. (a) What is a perfectly balanced binary tree? (3)

   (b) What is a depth (height) of a tree? (3)

   (c) How many nodes does a perfectly balanced binary tree of depth $d$ contain? (4)

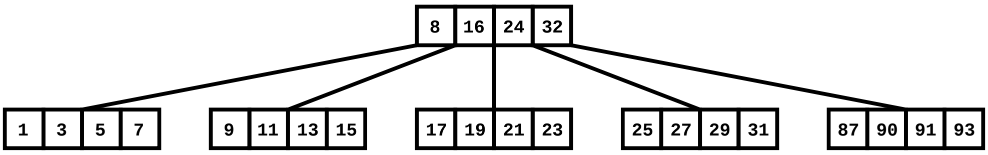
   (d) Give an inductive proof of the answer to question 2(c). (15)

3. (a) What is a B-tree? (3)

   (b) Where are B-trees used and why? (6)

   (c) Explain how insertion in B-trees works (in English or pseudocode, you may also use pictures). (10)

   (d) Draw the result of inserting an item with key 33 in the following B-tree of order 5: (6)



4. (a) Write in Java or pseudocode a linear time algorithm for sorting an array of positive integers which relies on the fact that all integers in the array are less than some integer k. Assume that all numbers in the array are different. (10)

   (b) Do you need to change your algorithm if the array may contain repetitions? If yes, describe the changes. (5)

   (c) Prove that the worst-case complexity of your algorithm is O(N). (5)

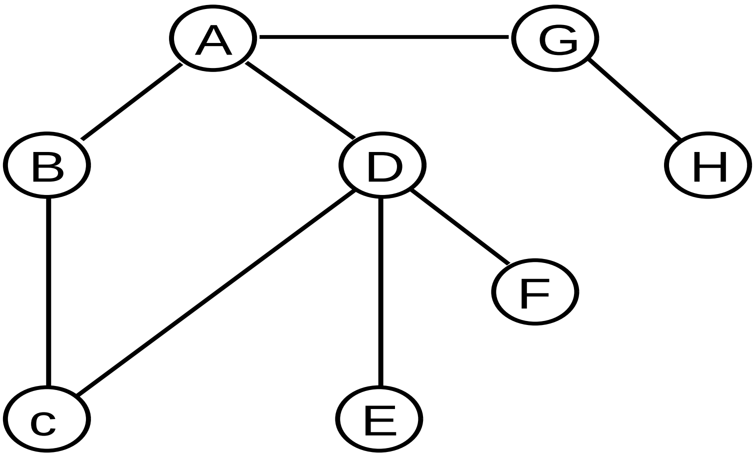
   (d) What is the space complexity of your algorithm? (5)

5. (a) How would you implement a graph data structure (give Java code or pseudocode). Illustrate on the example of the graph below (question 5(d)). (5)

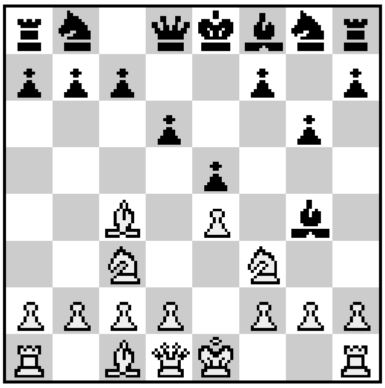
(b) Give Java code or pseudocode for depth first search. (10)

(c) What is the worst-case time complexity of your implementation? (5)

(d) Trace depth first search on the following graph starting at node A: (5)



6. Suppose you are writing a chess-playing program. One of the components of the program is an evaluation function for (some) positions in the game which returns the probability of a win in this position for each player. This information will be needed during the game and should be accessible very fast. You already have the information but it is stored in a format unsuitable for use by the program, for example, in a book where each position is represented by a picture with the corresponding probabilities: **P(white)=0.4, P(black)=0.55**.



   (a) Which data structure will you use to store information about positions. Describe in detail how the data will be stored and retrieved. (15)

   (b) Justify your choice of the data structure. Analyse the efficiency of your implementation, giving big-Oh complexity in the worst and average case for the retrieval method. (10)

**Background information.** Chess is a two player game. One player plays with white figures and the other with black ones. The board consists of 8 × 8 squares numbered *a* to *h* and 1 to 8. Initially each player has eight Pawns, a King, a Queen, two Rooks, two Knights, and two Bishops on the board. In the course of the game some Pawns may turn into Queens, and any number of figures apart from the two Kings may disappear. You may assume that any arrangement of figures on the board is possible. You don't need to know anything else about the rules of chess to be able to answer this question.