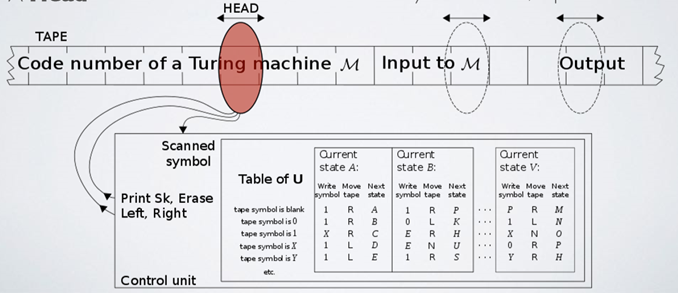
2017-18 Exam Paper

1)

a. Describe the operation of a Turing machine. In your answer you should consider the roles of the tape, symbols, states and operations. (8 marks)

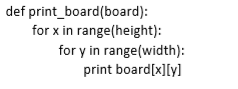
* The Turing Machine uses an inﬁnite tape as its unlimited memory.
* It has a tape head that can read and write symbols and move around on the tape.
* Initially the tape contains only an input string and is otherwise blank ( ).
* If the machine needs to store information then it writes this information on the tape.
* To read the information it has written, the machine can move its head back over it. The machine continues computing until it decides to produce an output. The outputs accept and reject are obtained by entering designated accepting and rejecting states.
* if it doesn’t enter an accepting or a rejecting state, then it will go on forever, never halting

A Turing machine is an abstract machine that manipulates symbols on a strip of tape according to a set of rules. The Turing machine uses a tape of infinite length, A state register, A finite table of instructions, A head. Firstly, the Turing machine reads the tape then decodes that information on the tape, then if necessary it will act on any instructions that it is given based on the table of instructions.

b. What is meant when Computer scientists refer to the halting problem? In your answer you should briefly describe the problem and explain the implications for everyday computer use. (8 Marks)

The Halting problem refers to determining from a description of an arbitrary program and input whether the program will finish running or continue. In the context of Turing machines, given a description of a Turing machine and initial input, asks whether the program when executed on the input will halt (complete) or continue forever. Its an example of an undecidable or non-computable problem. The main implication that it has for everyday computer use is that sometimes a problem is non programable or computable with the current programming languages we possess. Also the only way to know if given a program will halt on a particular input in all cases is to simply run it and see if it halts. If it does halt then you know it halts. If it doesn’t halt however, you may never know if it will eventually halt.

c. You have implemented a game that is played on a board that is x squares wide by y squares high. To print the contents of every board location you have the following function:



Using “Big O” notation describe the complexity of this function and explain what happens as boards of increasing size are used. ( 9 Marks)

Big Oh refers to the “order” associated with the performance, i.e. the degree of complexity, so O(n) is read “The order of n”. Essentially we are approximating orders of magnitude, i.e. Does the algorithm run in constant time, linear time, quadratic time, logarithmic time?, This lets us predict how a given algorithm will perform given input size.

The complexity of this function is O(n) as it runs in linear time (If the time it takes to execute is directly proportional to input size) because the bigger the board you have the more values that must be accessed and returned within the function.

2)

1. Describe with an example, what is meant by the phrase “divide and conquer algorithms”. (5 marks)

A divide and conquer algorithm is a problem solving algorithm that uses the following 3 steps.

Divide – Break the given problem into subproblems of the same type.

Conquer – Recursively solve these subproblems

Combine: Appropriately combine the answers.

Some examples of Divide and conquer algorithms are Binary Search, Quicksort, Merge Sort.

1. Using basic data structures, outline a strategy for identifying whether a given string is a Palindrome (a string that reads the same backwards as forwards, for example “Madam”). In your answer you should consider any limitations to your approach.

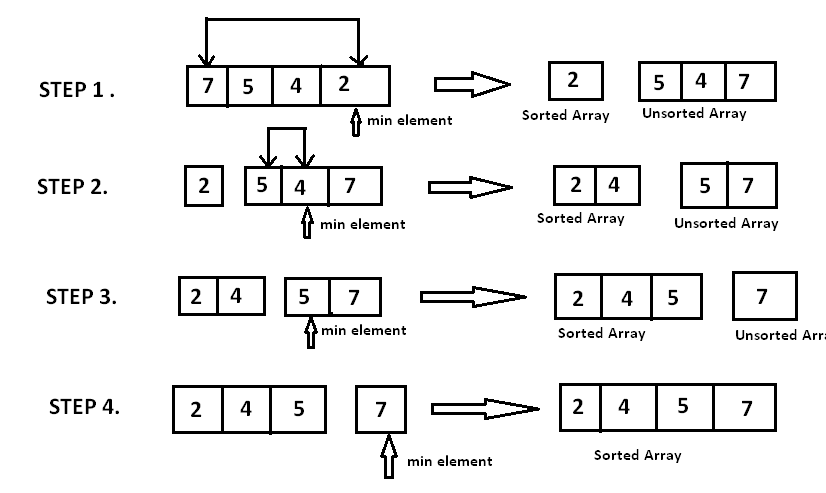
(8 marks)

To start this problem, I would have two strings one with original value (the input) and a string output with the string reversed. To do this I would pass the input string into a function that would pass the input string into the a char[] array. I would then use a loop to swap each character at the start of the string with its corresponding character at the end of the string (e.g. The first value with the last, the second with second last etc). I would then pass this into the output string and compare the input and output and if they are the same then return that the string is a palindrome and if not then return that it is not a palindrome). The limitations of this approach would be that it requires several data structures which each require their own amount of memory. Also, if the input string is very large the time complexity will increase as it has a O(n) time complexity.

1. Apply the selection sort algorithm to the following array of numbers:



For each iteration of the algorithm identify the least element in the unsorted list and write out the sorted and unsorted sub-lists. (12 marks)

Example Answer

Iteration 1 – Least element = 17, Sorted array = [17], Unsorted array = [21,86,78,29,76]

Iteration 2 – Least element = 21, Sorted array = [17,21], Unsorted array = [86,78,29,76]

Iteration 3 – Least Element = 29, Sorted array = [17,21,29], Unsorted array = [86,78,76]

Iteration 4 – Least Element = 76, Sorted array = [17,21,29,76], Unsorted array = [86,78]

Iteration 5 – Least Element = 78, Sorted array = [17,21,29,76,78], Unsorted array = [86]

Iteration 6 – Least Element = 86, Sorted array = [17,21,29,76,78,86], Unsorted array = []

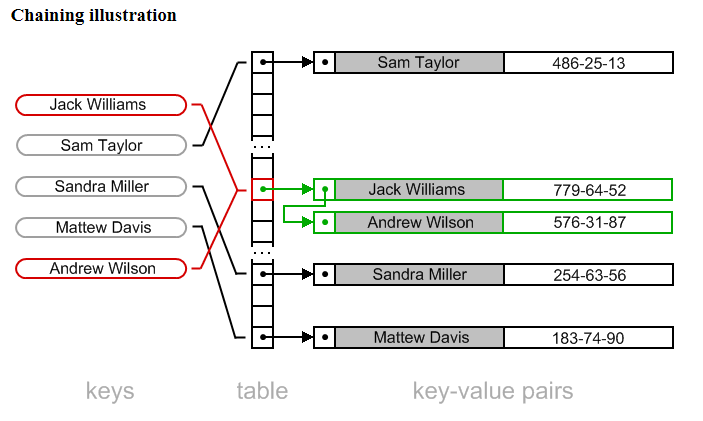
3) Consider a chained hash table of size 5, where each bucket is implemented as a linked list where new elements are inserted at the head. The hash table uses the following hash function:



1. Insert the following elements in order, showing the array and linked lists after inserting each element:



Chaining is a possible way to resolve collisions. Each slot of the array contains a link to a singly-linked list containing key-value pairs with the same hash. New key-value pairs are added to the end of the list. Lookup algorithm searches through the list to find matching key. Initially table slots contain nulls. List is being created, when value with the certain hash is added for the first time.



1. Explain the relationship between the load factor of the hash table and the time taken to find an item in a chained hash table. (4 marks)

If the load factor is low, then the capacity of the hash table will increase quicker than if the load factor was high as it will increase the capacity sooner or by more compared to a high load factor. So if the load factor is low the time take to find an item in a chained hash table is slower than that of a high load factor.

1. Give one advantage of chained hash tables compared to open-addressed hash tables

(2 marks)

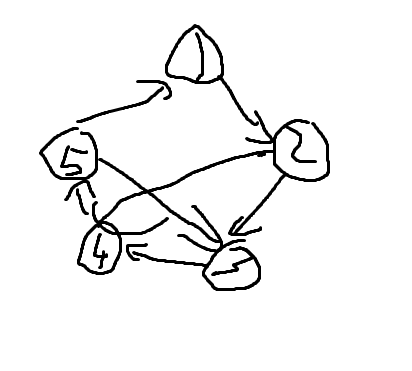
One advantage of chained hash tables over open addressed hash tables is that deletion is easier in chained hash tables.

1. Passwords are typically stored on a server in hashed form rather than as plain text. Explain how adding a salt to a password before hashing helps against rainbow table attacks. ( 5 Marks)

Salt helps if the attacker wants to break many passwords. This is usually the case. Sometimes the attacker is attacking a site and wants to break into an account on that site, any account. Without a salt, most of the attacker's work can be used for all accounts, so she can test each of her attempts against all accounts at once. With a correctly-chosen salt (i.e. if no two accounts have the same salt), the attacker must start over for each hashed password.

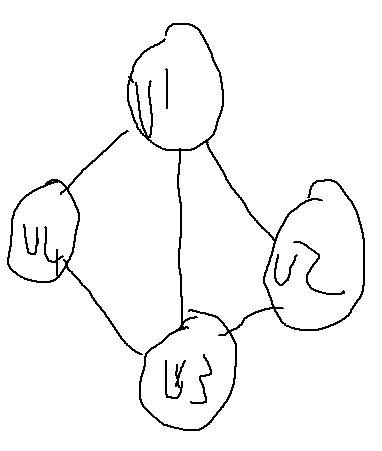
4)

a. Draw a directed graph with five vertices and seven edges. Exactly one of the edges should be a loop and do not have any multiple edges. (5 Marks)

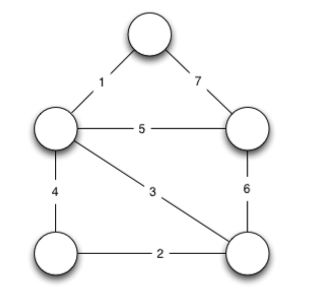


b. Draw an undirect graph with four vertices and five edges. The vertices should be called v1, v2, v3 and v4 – and there must be a path of length three from v1 to v4, indicate the path from v1 to v4.

(5 Marks)



1. A telephone company plans to connect its switching centres together using fibre optics. When connecting switching centres, it is required to have a path between every pair of them. The telephone company is looking for an interconnection topology that minimizes the amount of fibre required to connect its switching centres. The switching centres are connected by pipelines along which the fibre can be laid according to the diagram below:



1. Which algorithm can be used to help the telephone company find the optimal topology.