ALGORITHMICS UNIT 3 & 4   2022

Trial Exam 1: 2022 SOLUTIONS

Section A: Multiple Choice Answers

Question 1 A	Question 6 C	Question 11 D	Question 16 A
Question 2 A	Question 7 A	Question 12 A	Question 17 B
Question 3 D	Question 8 A	Question 13 D	Question 18 B
Question 4 C	Question 9 D	Question 14 A	Question 19 C
Question 5 B	Question 10 D	Question 15 A	Question 20 A

SECTION B – Extended Response Questions Answer all questions in the space provided.

Question 1 (12 marks)

a. Concisely contrast Breadth First Search (BFS) and Depth First Search (DFS). (2 marks)
BFS visits nodes in "layers": those one step away from the source are visited, then all those two steps away, etc. DFS starts at the source and goes as far as possible. When it hits a dead end, it backs up to the predecessor and goes as far as possible from there. This repeats until the algorithm backs up to the source. (Illustrations are also helpful here, as would be speaking of broad vs. deep spanning trees.)

b. An elementary data structure is used in BFS. Which one, how is it used (2 marks)
A queue. It holds unprocessed vertices, which are those we have discovered but whose adjacency lists we have not yet fully explored. All vertices one layer from the source get enqueued first, then all vertices two layers away, etc. So, when dequeuing happens, as it does in FIFO style, vertices come off the queue in layers.

c. how much time is spent adding and removing data to the data structure (2 marks)
Each vertex gets added to the queue once and each vertex gets dequeued once. This makes for a total of 2|V(G)| operations [if G is connected] or O(|V(G)|) [if G is not connected, as we don't get to all vertices], asymptotically

d. Apply BFS with A as the start node (2 marks)

A, B, D, E, C, F	queue (adjacent nodes put in queue if parent node is not already processed) A B D E D E D E D C F D C F D C F D F

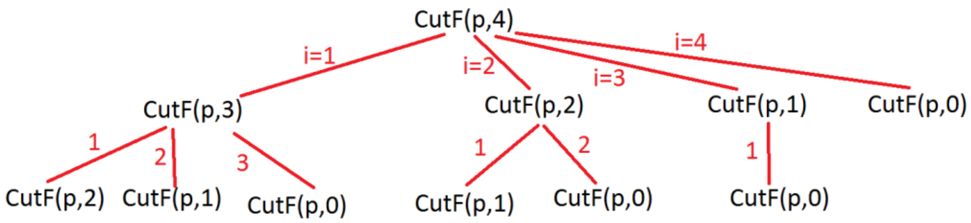
e. Apply DFS with A as the start node. (2 marks)

A, B, D, C, F, E	Stack (adjacent nodes put in queue if parent node is not already processed) A B D E D D E C F D E B F D E F D E D E E

f. Explain Best first search comparison to BFS and DFS (2 marks)
Best First Search uses a priority queue to expand the path according to some optimal goal, the nodes in the priority queue are ranked by some heuristic that tries to optimise a path from the source to the target. Breadth First Search and Depth First search do not have an optimal path goal.

Question 2 (12 marks)

a. call tree that will result when the above Function with n=4. (2 marks)



b. What is the recurrence relation describing the recursive call as a function of the input size n? (1 mark)

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

c. Hence determine the time complexity of the above Function. (2 marks)

By telescoping $T(n) = nT(n-1) + 1$ $T(n) - nT(n-1) = 1$ $T(n-1) - (n-1)T(n-2) = 1$ $T(n-2) - (n-2)T(n-3) = 1$ :: $T(2) - 2T(1) = 1$ $T(1) - T(0) = 1$	$T(n) - nT(n-1) = 1$ $(n)T(n-1) - (n)(n-1)T(n-2) = (n)1$ $(n)(n-1)T(n-2) - (n)(n-1)(n-2)T(n-3) = (n)(n-1)1$ :: $n!\,T(1) - n!\,T(0) = n!$

By elimination in the telescoping $T(n) = n!\,T(1) + n! + \frac{n!}{2!} + \frac{n!}{3!} + \cdots$.. which is O(n!)

d. Describe how the time complexity of this problem be improved using Dynamic Programming. (2 marks)

- Problem with naïve recursive solution: sub-problems solved multiple times, DP solves each sub-problem just once by building its solutionBottom and store solution to subproblems of length i in array element r(i)
- DP Used when: • Optimal substructure • Overlapping subproblems Methodology • Characterize structure of optimal solution • Recursively define value of optimal solution • Compute in a bottom-up manner

e. Complete the following Function that uses Dynamic Programming to find the solution to the problem for the general case. (3 marks)

```
   for j in 1 .. n loop
        q := -1000000
        for i in 1 .. j loop
             // Find the max cut position for length j
             q := maximum(q, p[i] + r[j-i])
        end loop
        r[j] := q
   end loop
   return r[n]

end Function
```

f. What is the time complexity pf DPCutF? Justify your answer. (2 marks)

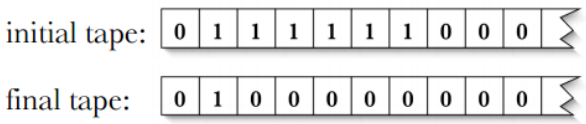
Two nested loops (n)(n) $O(n^2)$

Question 3 (8 marks)

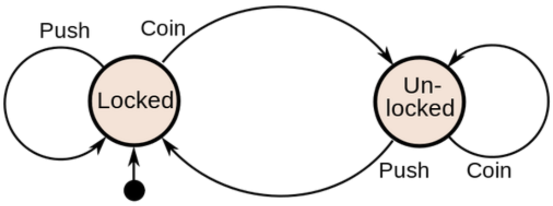
a. Describe the components of a Turing machine and how they function (2 marks)

Unlimited tape with symbols on it, an input/output device to read/write to tape, a table that defines actions based on current state and input from tape.

b. final tape that will be returned by the Turing machine described above. (2 marks)

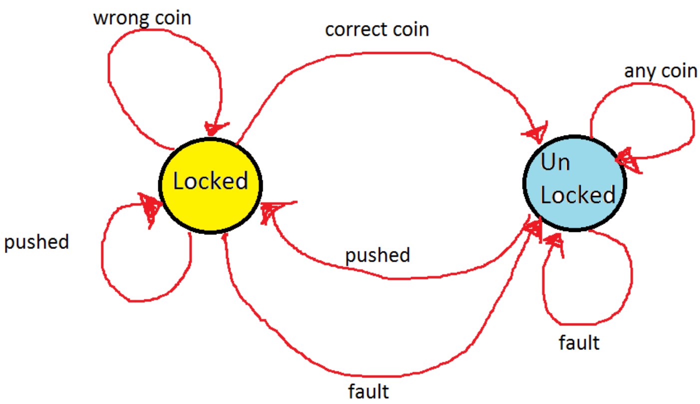


c. Show the information for the Turnstile FSM as a connected graph. (2 marks)



Answer

a. Create a new transition graph depicting the FSM to handle the existing and the extra cases. (2 marks)



Question 4 (11 marks)

a. What was Hilbert's program and what prompted Hilbert to initiate a program to make Mathematics a system that was complete, consistent and decidable? (2 marks)

- The Mathematician Hilbert thought the field of mathematics was not built on sturdy foundations and had many ambiguities. Hilbert's formalism was premised on the idea that the ultimate base of mathematics lies, not in logic itself, but in a simpler system of pre-logical symbols which can be collected together in strings or axioms and manipulated according to a set of "rules of inference".
- His ambitious program to find a complete and consistent set of axiomatic proof system for all of mathematics.

b. What are the implications of Godel's incompleteness theorem, how does it impact Hilbert's goal of a complete, consistent and decidable Mathematical language and system? (3 marks)

- Gödel's incompleteness theorem showed that, within any logical system for mathematics that mathematicians ever come up with will always rest on at least a few unprovable assumptions.
- A finitary system cannot be used consistently to prove itself. Consistency and completeness were not possible using finitary axioms
- Godel's incompleteness theorem implies that not all mathematical questions are even computable using a finitary system.

c. What are the implications of Turing's Halting problem for the creation of algorithms to solve problems? How does it impact Hilbert's goal of a complete, consistent and decidable Mathematical language and system? (3 marks)

- Turing recast decidability in terms of a Turing machine. He first proved that such a machine would be capable of performing any conceivable mathematical computation if it were representable as an algorithm.
- He then went on to show that, even for such a logical machine, essentially driven by arithmetic, there would always be some problems they would never be able to solve, that would be undecidable, and that a machine fed such a problem would never stop trying to solve it, but would never succeed (known as the "halting problem").
- Turing proved that there was no way of telling beforehand which problems were the unprovable ones, or undecidable for any input.

d. Explain how Hilbert's program contributed to the foundations of Computer Science and to the definition of algorithms and what is computable and decidable? (3 marks)

- Hilbert's program initiated a lot of research and exploration of how to go about proving a course of action described by logical steps could achieve a required Mathematical outcome or proof, which is the decidability goal.
- To find if Hilbert's goal of decidability could be achieved Turing and Church had to independently define what an algorithm was by defining them on by a Turing Machine and Lambda calculus
- This formed the basis of algorithmic thinking and the foundation of Computer Science with the development of the Church Turing Thesis that defines computable functions as those that can be solved on Turing Machine or their equivalent Lambda Calculus.

Question 5 (10 marks)

a. Using the following directed graph defining symbols to be used in order for correct sentences, complete the Algorithm CheckSentence that will check any sentence made up of a list of words for correctness using a directed graph with a nominated starting and end point. (6 marks)

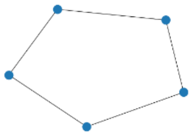
```
//    Assume start node is labelled 1
//    and each node of syntax is labelled in order 1,2,3….n
//    Edges have permitted words as labels
      For i:=1 to Diam do
             Match:=false
             For each outgoing edge e of node[i] do
                     If (Word[i] == e) then
                              Match:=true
                     End if
             End Loop
             If (Match is false) then
                     Report "Incorrect syntax"
                     Exit
             End if
        End Loop
End if

Report "Sentence is ok"
End Algorithm
```

b. What other abstract datatypes could be used to support the words and order of use of "correct sentences"? Justify your choice and give an example of how it would be used to support the information in the connected graph above. (4 marks)

- Many possible ADTs can be combined for a correct solution
- For example a queue could maintain the order of syntax
- Options for each queue item could be maintained in a list

Question 6 (7 Marks)

A round graph is a graph with at **least three nodes** and in which a single cycle exists containing all vertices. An example of a ring graph with five vertices is shown below.



Consider the algorithm shown below with each line numbered in the left column, this algorithm accepts one integer input n.

```
1   Algorithm RoundGraph(n)
2   // Assumption n is at least 3
3       If (n=3) then
4           Return a complete graph with three nodes
5       Else
6           G = RoundGraph(n-1)
7           Add new node u to G
8           Randomly select edge p-q in G
9           Create edge p-u and u-q to G
10          Delete the edge p-q
11          Return G
12      End if
13  End Algorithm
```

a) Show the call graph for RoundGraph(5). (1 mark)

RoundGraph(5)→RoundGraph(4)→RoundGraph(3) base case reached

b) What are the design pattern(s) used for the RoundGraph algorithm? Justify your response. (2 marks)

Recursive, Decrease and Conquer algorithm design patterns are used, on line 6 of the algorithm RoundGraph calls itself and decreases the size of the input by 1.

c) Demonstrate the correctness of the RoundGraph algorithm, the numbered lines of the algorithm and diagrams can be used to support the proof. (4 marks)

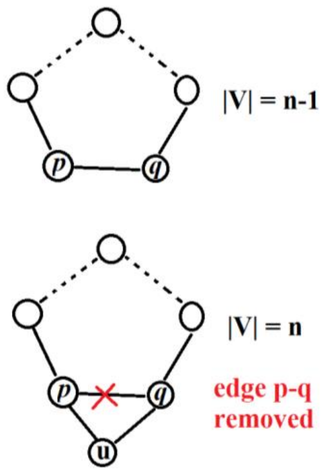
Inductive hypothesis: RoundGraph returns a round graph for $n \geq 3$

Base case: When $n = 3$, the algorithm returns a complete graph with three nodes, which is a round graph.

Inductive step: Assume that RoundGraph(n-1) returns a ring graph.

RoundGraph(n) begins with the result of RoundGraph(n-1).

In lines 8 – 10, a new node u is added to graph G, which now has n nodes, and edge (p,q) in G is replaced with a new edges (p-u) and (u-q). This replaces one path from p to q with one new path from p to q going via an intermediate node u and hence the graph remains a round graph. Therefore, if RoundGraph(n-1) returns a round graph for n-1 nodes, then RoundGraph(n) returns a round graph of n nodes. As RoundGraph(3) is correct and RoundGraph(n) is true if RoundGraph(n-1) is true, then RoundGraph(n) is true for $n \geq 3$.



Question 7 (8 Marks)

<u>The Restaurant Problem</u>

You are the organiser for a Youth United Nations Conference, it is the first night and you've forgotten to book a restaurant. The rules for the first night dinner is that **<u>no</u>** two delegates from the same country can eat at the same restaurant. Due to the limited seating at some restaurants, this arrangement may not be possible and some delegates may not be able to eat at a restaurant. You would like to minimise the number of delegates that miss out so as not to cause an international incident.

Algorithm Input Variables to be used to answer part b)	**Algorithm Output**
• Integer **C** – the number of countries attending the conference • Input **CList** a list of countries attending • Input **DCList** is list of number of delegates attending from each corresponding country in **CList** • Integer **R** – the number of available restaurants • Input **SList** – giving the number of seats available at each restaurant	Your output must consist of a single value containing a single integer, the total number of delegates that cannot be seated at a restaurant.

Sample Input 1	Output 1	Explanation
C=2 DCList={3,3} R=3 SList={4,3,4}	0	The delegates are easy to seat in this case--simply place one member of each team in each restaurant. Since every restaurant has more than two seats available, everybody can be seated and nobody misses out.
Sample Input 2	**Output 2**	**Explanation**
C=3 DCList={4,3,3} R=3 SList={5,2,3}	2	Here the first country has more delegates than there are restaurants, so at least one of its delegates must miss out. The second restaurant only has space for delegates from two countries, and so a second delegate must miss out. It is possible however to arrange the seating so that only two delegates miss out overall, and so the final solution is two.

a. **<u>The Restaurant Algorithm (6 marks)</u>**

```
Function Restaurant(C, CList, DCList, R, SList)
   For i=1 to C loop
         For j = 1 to R loop
              If (SList[j] > 0) and (DCList[i] > 0) then
                  //Seats are available and Delegates still need to be seated
                  DCList[i]:=DCList[i] -1   // seat the Delegate
                  SList[j] := SList[j] – 1  // seat is allocated
              End if
         End Loop
   End Loop
   //  DCList will now contain those who couldn't be seated
   Notseated :=0
   For i=1 to C loop
             Notseated := Notseated + DCList[i]
   End loop
   Return NotSeated
End Function
```

b. How can graph ADTs be used to model this problem? Explain the strategy. (2 marks)

Problem can be modelled with Graph ADT to show conflict, the nodes could be the restaurants and the edges can represent conflict of countries, graph colouring problem solution can be created.

Question 8 (12 marks)

a) Can Neural Networks after supervised training be categorized as "Weak" or "Strong" AI? Explain and justify your responses. (2 marks)

This is weak AI as no understanding is developed by Neural Network during its training, the input information has been correlated by the Neural Network to the required outputs by adjusting weights on the inputs and applying mathematical functions.

b) The philosopher Searle believes that Artificial Intelligence is not possible. Describe the thought experiment he devised to demonstrate that Artificial Intelligence is not possible. (2 marks)

- "Suppose that I'm locked in a room and ... that I know no Chinese, either written or spoken".
- suppose that I have a set of rules in English that "enable me to correlate one set of formal symbols with another set of formal symbols", that is, the Chinese characters.
- These rules allow responses, in written Chinese, to questions, also written in Chinese, in such a way that the posers of the questions – who do understand Chinese – are convinced that Searle can actually understand the Chinese conversation too, even though he cannot.

c) Complete the table below describing "The Robot Reply" and "The Brain Simulator Reply" to Searle's argument together with Searle's response to the objections. (8 marks)

Robot Reply	**Searle's response to Robot Replay**
A fixed computer (or person) as confined to the room does not understand Chinese under present considerations. If it had sensory apparatus to connect words with meaning and was suitably programmed then it would be able to understand Chinese, forming a connection to the symbols from the realities they are supposed to represent, to promote the "symbol" manipulation to genuine understanding.	"some of the Chinese symbols come from a television camera attached" Still, Searle asserts, "I don't understand anything except the rules for symbol manipulation." He explains, "by instantiating the program I have no [mental] states of the relevant [meaningful, or intentional] type. All I do is follow formal instructions about manipulating formal symbols." Searle also charges that the robot reply "tacitly concedes that cognition is not solely a matter of formal symbol manipulation" after all, as "strong AI" supposes, since it "adds a set of causal relation[s] to the outside world"
Brain Simulator Reply	**Searle's response to Simulator Reply**
if the program simulates the actual sequence of neuron firings at the synapses of a Chinese speaker when he understands stories in Chinese and gives answers to them." Surely then "we would have to say that the machine understood the stories"; or else we would "also have to deny that native Chinese speakers understood the stories" since there would be no difference between "the program of the computer and the Chinese person's brain"	If instead of symbols we "have the man operate an elaborate set of water pipes with valves connecting them, where each water connection corresponds to synapse in the Chinese brain." Given some Chinese symbols as input, the program now tells the man "which valves he has to turn off and on. … so that after . . . turning on all the right faucets, the Chinese answer pops out at the output end of the series of pipes." Yet, Searle thinks, obviously, "the man certainly doesn't understand Chinese, and neither do the water pipes." Searle diagnoses the insufficiency of this formal structure for producing meaning and mental states.